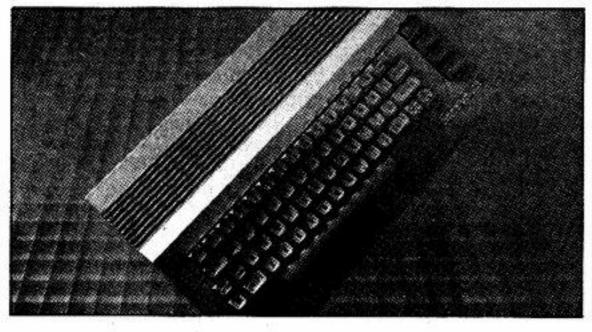
Sommario

Foto di copertina di Sergio Ferrero

Campus 64 / 128

Amigames

Da pagina 41 a pagina 56 le ultimissime novità che stanno arrivando in questi giorni in Italia anche attraverso vie poco... ortodosse

Campus Amiga

Usa il tuo computer

Rubriche

C.C.C: le prossime scadenze

Dal mese di **luglio '90** verranno limitate le collaborazioni per il C/128.

Dal **gennaio '91** verrà evasa prevalentemente la corrispondenza pervenuta a mezzo BBS (modem).

Dal **gennaio '91** verranno accettate le collaborazioni che perverranno in Redazione a mezzo BBS.

Dal **gennaio '91** verranno affrontati argomenti relativi solo al C/64 e Amiga.

Dal **dicembre '92** verranno affrontati argomenti relativi solo ad Amiga e sistemi Ms-Dos.

Commodore Computer Club

COMMODORE COMPUTER CLUB

Direttore: Alessandro de Simone
Coordinatore: Marco Miotti

Redazione / Collaboratori:
Davide Ardizzone - Claudio Baiocchi
Luigi Callegari - Umbero Colapicchioni
Donato De Luca - Carlo D'Ippolito
Valerio Ferri - Michele Maggi
Giancarlo Mariani - Domenico Pavone
Armando Sforzi - Dario Pistella
Fabio Sorgato- Valentino Spataro
Franco Rodella- Stefano Simonelli
Luca Viola

Grafica: Arturo Ciaglia

Direzione:
Via Mosè, 22 cap. 20090 OPERA (Mi)

Telefono 02 / 55.50.03.10
Fax 02 / 57.60.30.39
BBS 02 / 52.49.211

Pubblicità:
Leandro Nencioni (dir. vendite)
Via Mose', 22 20090 Opera (Mi)
tel. 02 / 55.50.03.10

Emilia Romagna:
Spazio E
P.zza Roosvelt, 4 cap. 40123 Bologna
Tel. 051 / 23.69.79

Toscana, Marche, Umbria
Mercurio s.r.l. Via Rodari, 9
S. G.nni Valdarno (Ar)
Tel. 055 / 94.74.44

Lazio, Campania
Spazio Nuovo
Via P. Foscari, 70
cap. 00139 Roma
tel. 06 / 81.09.679

Abbonamenti: Liliana Spina
Arretrati e s/w: Lucia Dominoni

Tariffe: Prezzo per copia L. 6000
Abbonamento annuo (11 fascicoli) L. 50000
Estero: L. 100000 - Indirizzare versamenti a:
Systems Editoriale Srl c/c 37952207 oppure
inviare comune assegno bancario non
trasferibile e barrato due volte a:
Systems Editoriale Srl (servizio arretrati)
Via Mose', 22
cap. 20090 OPERA (Mi)

Composizione: Systems Editoriale
La Litografica Srl Busto Arsizio (Va)

Registrazioni: Tribunale di Milano
n. 370 del 2/10/82

Direttore Responsabile: Michele Di Pisa

Spedizioni in abbonamento postale gruppo
III. Pubblicita' inferiore al 70%

Distributore: Parrini - Milano

Periodici Systems: Banca Oggi -
Commodore Club (disco) - Commodore
Computer Club - Commodore Computer
Club (disco, produzione tedesca) - Computer
- Computer disco - Electronic Mass Media
Age - Energy Manager - Hospital
Management - Jonathan - Nursing '90 - PC
Programm (disco) - Personal Computer -
Security - Software Club (cassetta ed.
italiana) - TuttoGatto - Videoteca
VR Videoregistrare

Editoriale

Un po' per volta il nuovo percorso di Commodore Computer Club, indicato a partire da dicembre '89, sta prendendo una fisionomia ben precisa.

I lettori più attenti avranno certamente compreso lo scopo del brutale cambiamento di rotta, all'inizio, ed il graduale spostamento verso i consueti orizzonti che da sempre ci hanno caratterizzato.

La "cattura" degli utenti che pensavano solo a giocare (non immaginando nemmeno che è anche possibile divertirsi programmando) ha consentito di poggiare su consensi più ampi la solidità della nostra pubblicazione, a tutto vantaggio dell'informatica dilettevole.

Il graduale abbandono dei videogames usa e getta, "criticati" senza pietà dai nostri collaboratori, ha felicemente insinuato, nei pensieri dei neo-utenti, l'idea che i videogiochi sono quasi tutti eguali tra loro e che, soprattutto, non vale la pena passare pomeriggi interi nel tentativo di annientare il maggior numero possibile di marzianetti.

La contemporanea "campagna" promozionale sull'adeguamento tecnologico degli impianti, benchè domestici (leggi: compratevi un modem, è il momento buono) ha superato le più ottimistiche previsioni, tanto che gli argomenti legati alla telecomunicazione saranno presenti più spesso sulle nostre pagine.

Qualcosa, inevitabilmente, doveva sacrificarsi; la sorte (o una certa negligenza della stessa Commodore?) ha voluto che la vittima designata fosse il C/128, la cui morte è stata preannunciata sul numero scorso. A chi toccherà la stessa sorte subita, in precedenti periodi, dal Pet, dal Vic 20, dal C/16, dal Plus 4, dal C/128 (modo Cp/m) e dall'Amiga 1000?

Nessuno è in grado di prevedere il futuro, nè esistono programmi sufficientemente sofisticati per provvedere alla bisogna. L'idea di preannunciare, con larghissimo anticipo, l'abbandono di determinati argomenti, tuttavia, è piaciuta a molti; ad altri un po' meno, ma c'era da aspettarselo.

Il tempo per sostituire i vecchi computer, quindi, c'è ed è più che sufficiente, anche tenendo conto di eventuali, inevitabili abbassamenti dei prezzi.

Ai lettori, quindi, il compito di riflettere adeguatamente; o di restare indietro nel tempo.

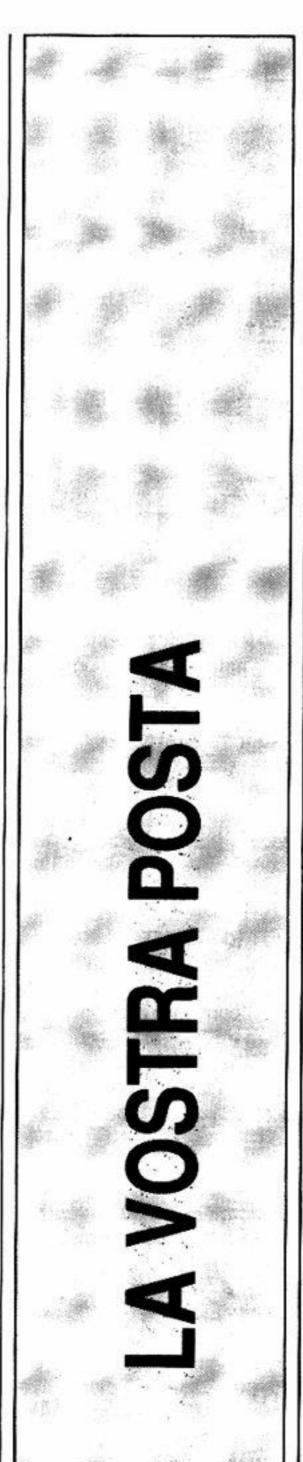
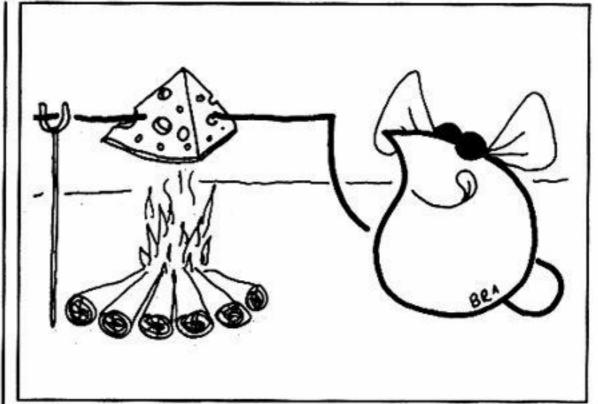
Lavare la testa

Esistono dischetti pulisci-testina per il drive 1541, e per i drive in genere?
(Massimo Cascella - Massafra)

Adoperando spesso i drive, come è giusto che sia, il materiale magnetico depositato sulla superficie del dischetto tende ad essere asportato per sfregamento. La testina di lettura / scrittura, infatti, preme sulla superficie del disco con una certa forza, necessaria per assicurare l'aderenza in grado di garantire il corretto flusso di informazioni. Se la pressione delle **pinze** della testina sono ben regolate, di solito non c'è bisogno di manutenzione straordinaria. Il drive del mio C/128-D, per esempio, non l'ho mai pulito da quando l'ho acquistato (1983).

Può capitare, però, che l'uso particolarmente intenso della periferica, o il suo utilizzo da parte di un'utenza non molto attenta, costringa a far ricorso a particolari prodotti che, simili ai puliscitestina dei registratori, provvedono a rimuovere la sporcizia dalle testine dei drive. Per provvedere alla pulizia sono in commercio numerosi prodotti, più o meno simili tra loro.

E' necessario intervenire, però, solo se si verificano numerosi errori di lettura e/o scrittura che potrebbero, del resto, esser dovuti anche a disallineamenti della testina.

Gli **Hard disk** (dischi rigidi), tanto per restare in argomento, funzionano su un principio del tutto simile ai floppy disk. In questo caso, però, le testine di lettura / scrittura sono molto vicine alla superficie magnetica del disco (alcuni micron), ma non la toccano. Proprio per questo motivo (e soprattutto perchè sono sigillati) non c'è pericolo che vi sia polvere da asportare e non è necessaria alcuna manutenzione: quando il computer segnala un errore di lettura / scrittura, però, vuol dire che si sono verificati grossi guai...

Campioni a metà

Perchè non pubblicate più i nomi dei lettori che inviano le soluzioni o le Poke "segrete" dei videogames?
(da alcune lettere)

La rubrica "**Campioni del Software**" era nata per sollecitare i più esperti, tra i nostri lettori, in grado di esplorare il software (dilettevole o professionale, protetto o libero che fosse) in modo da divulgare sofisticate tecniche di programmazione.

Su ogni numero, a partire da quello successivo al lancio dell'iniziativa, sono quindi comparsi **solo** gli articoli di una certa "consistenza", che non lasciavano dubbi sull'originalità della procedura descritta.

Altri lettori, invece, si sono semplicemente limitati a trascrivere (da altre riviste del settore) le Poke e le procedure da seguire per giungere al risultato richiesto.

E' ovvio che non è possibile tener conto di tali lettere.

6499, la storia infinita

Con il mio 6499, tentando disperatamente di collegarmi...
(vari lettori delusi)

Alt! Mettiamoci una buona volta in testa che il 6499, con i suoi miseri **300 baud**, non è più al passo con i tempi; lo era vari anni fa, quando un modem appena decente superava, e di parecchio, il mezzo milione di lire.

Oggi, quindi, è molto più economico procurarsi un **vero modem** (da utilizzare a 1200 / 2400 baud, pilotato da s/w serio) e risparmiare sulla bolletta del telefono. Se poi, oltre a cambiar modem, cambiate anche il computer, saranno in pochi a piangere.

Oroscopo

Alla rivista C.C.C. n. 73 era allegata la cassetta "Oroscopo" che, caricata con il mio C/128 in modo 128, se-

Duplicate l'emulatore!

Volendo partecipare anch'io alla ricerca di nuovi metodi per sconfiggere il software originale (e poterlo poi scopiazzare), ho scoperto, per puro caso, che il dischetto "Ms-Dos emulator & Gw-Basic Emulator" (da voi prodotto) è, per così dire, autocopiante in quanto l'utility Dikcopy, presente sullo stesso disco, è in grado di copiare l'intero dischetto su cui si trova registrata, fornendo, oltretutto, una copia del vostro stesso emulatore, perfettamente funzionante.
Tant'è vero che io stesso, da circa un mese, non uso più il dischetto originale, ma la copia "pirateggiata" e non ho mai riscontrato alcun malfunzionamento (potevate metterci almeno un virus...).
Molto probabilmente ciò che ho scoperto io lo sapevate già, ma attenti: come l'ho scopiazzato io, usando ciò che voi avete messo a disposizione, potrebbe venire in mente a qualcun altro, per scopi non proprio personali, ma soprattutto economici.
Vi ringrazio fin d'ora se avrete il coraggio di pubblicare questa mia lettera...

(Jonathan lo sprotettore)

Non solo pubblico la lettera (e in buona evidenza), ma addirittura invito coloro che ancora non fossero a conoscenza del trucchetto(!), ad effettuare copie di "sicurezza"(?).

Il problema della pirateria, mio caro lettore anonimo (a proposito, credevi di rischiare una denuncia se firmavi la lettera? Figuriamoci...) non si risolve nè inserendo virus, nè applicando protezioni più o meno inutili.

Il problema, a mio parere, è di tipo *culturale*; noi di C.C.C. crediamo fermamente che l'informatica rappresenti uno degli strumenti indispensabili per il progresso dell'umanità. Maggiore è il numero di persone che, in una società, possiede "conoscenze" (e non nel senso di "raccomandazioni"), maggiore è la probabilità che la società progredisca verso situazioni migliori.

C'è da considerare, ovviamente, che per lanciare messaggi (o divulgare programmi) è necessario dedicare tempo, pazienza e denaro.

Tuttavia noi riteniamo che, con la pubblicazione del dischetto in questione (e con il ricavato delle vendite) sia gli autori, sia la Systems Editoriale, siano stati ricompensati a sufficienza della fatica (e del rischio economico) sopportati. La possibilità di guadagnare di più ("costringendo" una maggior quantità di lettori a procurarsi il software originale da noi prodotto) si scontra, pertanto, con la nostra impostazione di cui ho riferito all'inizio: l'informatica si diffonde solo se i costi da sostenere sono bassi.

In alcune osservazioni, riportate qui e là su C.C.C, è possibile individuare una certa simpatia per i pirati (ma, per carità, solo per quelli "buoni" e non per quelli che fanno copie a scopi di lucro).

I pirati "cattivi", però, vengono attirati esclusivamente dagli elevati introiti che è possibile ricavare divulgando software costoso. Ad esempio, un D.T.P. originale posto in vendita a due milioni può essere allegramente rivenduto, copiato, a centomila lire; il risparmio è più che evidente. Ma mi sai dire a che prezzo potrebbe esser rivenduto un software, pirateggiato, che, in originale, costa pochi biglietti da mille?

Il risultato è sotto gli occhi di tutti; nell'ambito degli utenti con cui ho avuto modo di parlare, ad esempio, non ne ho mai incontrato uno che abbia detto "posseggo il tal software originale" se il s/w in questione supera il prezzo di 150 mila lire.

E mi domando e dico: proponendo a L. 150 mila un programma, si vendono 100 copie (le altre si diffondono sul mercato della pirateria, inutile illudersi). Il ricavo totale è di 15 milioni.

Se offrono lo stesso programma a L. 50 mila, riescono a vendere (molto probabilmente) 500 copie, sottraendo una consistente fetta alla pirateria (ed il ricavo è di 25 milioni). Anche ammettendo che le spese per il maggior numero di copie (confezione, dischetti, libretti di istruzioni) raggiungono i 10 milioni, il guadagno è identico nei due casi. I pirati, però, subirebbero un duro attacco dal momento che non sarebbe più conveniente nè duplicare (da parte dei pirati) nè acquistare (da parte degli utenti finali). Inoltre (vuoi mettere?) la soddisfazione sarebbe maggiore.

Ed è proprio questo che, da sempre, è il motto vincente con cui vengono divulgati i programmi della Systems Editoriale: buona qualità, prezzo irrisorio.

gnalava diversi errori di sintassi...
(Gualtiero Miletto - Roma)

Il programma deve esser caricato sul C/128, ma in modalità 64. Pertanto, dopo aver acceso il C/128, digita...
Go 64
...(e **non** Goto 64). Alla domanda **"Are you sure?"** rispondi **"Y"** e premi il tasto Return. A questo punto effettua il caricamento come al solito.

Espansione 1764

L'espansione di memoria 1764 per C/64 è idonea per caricare, e far girare, programmi in Turbo Pascal (sistema Ms-Dos) dal momento che porta la memoria del C/64 a 256 Kram?
(Claudio Portelli - A. Piceno)
(A. Bonadio - Corigliano)

No. Anzi, sarò più cattivo: per sfruttare l'ingombrante espansione citata è necessaria una certa esperienza di programmazione l.m. dal momento che è indispensabile saper giocare bene con puntatori, banchi di memoria e simili piacevolezze. In definitiva, grazie al proverbiale disinteresse della casa madre, il software sviluppato su C/64 dotato di espansione può girare solo se scritto in linguaggio macchina.

Per ciò che riguarda il secondo lettore, la domanda sul

Suggerimenti

Avrei da fare una richiestina alla Commodore. Parlando di Amigabasic sarebbe interessante effettuare un paio di modifiche (anzi 3):
1) Attualmente Amigabasic usa ,come font caratteri, sempre e solo il topaz, anche se il sistema (ad esempio con setfont) ne ha caricato un altro. Sarebbe utile modificarlo in modo da visualizzarere il font che trova "attualmente" in uso. Ad esempio, se da Cli, con Setfont, si carica l'IBM.FONT, di facilissima reperibilità, se anche il basic che lo trova attivo lo usasse renderebbe le cose più semplici quando si usano programmi importati dal gwbasic.
2) Il comando CLEAR permette di assegnare al basic la quantità di ram necessaria. Ormai la quasi totalità di amiga ha almeno 1 mega di ram! Sarebbe più comodo se il default fosse almeno 50K, al posto del 25K attuali.
3) C'è in giro, per le bbs, una routinetta basic che permette ad amigabasic di accedere a Cli, ma è piuttosto lunga e, se non bastasse, si deve "accodare" al programma che la utilizzerà. L'ideale sarebbe aggiungere ad amigabasic un comando del tipo..."CLI". Sarebbe bello, nel bel mezzo di un programma, impartire...
CLI "show nomeschermata"
...per far visualizzare una schermata HAM.

(Alan Primo)

1700/1750 non è affatto chiara e non mi è possibile rispondere.

A proposito, permettetemi di approfittare della mia posizione privilegiata per la pubblicazione del seguente annuncio:

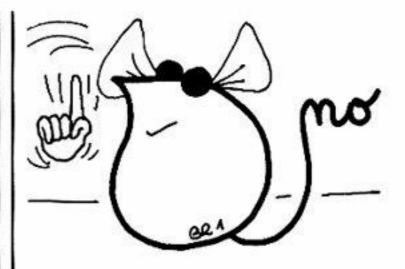
Causa inutilizzo, vendo (a sole novantamila lire) espansione di memoria per C/64, completa di alimentatore supplementare, disco demo e libretto di istruzione in inglese. Telefonare in Redazione (Commodore Computer Club) o lasciare messaggio in BBS.

Il telefonooo!

Verrei proporre alcuni fumetti che ritengo "in linea" con la vostra pubblicazione.
(A. Paolini - Pomezia)
F. Risatti - Limone)

Vi supplico in ginocchio(!) e ciò vale anche per altri aspiranti collaboratori: quando inviate una proposta, indicate anche il vostro **recapito telefonico**. Credetemi, è praticamente impossibile trovare il tempo per rispondervi per iscritto, nè posso sottrarre spazio alla rivista per spiegare in che modo sviluppare una collaborazione così particolare.

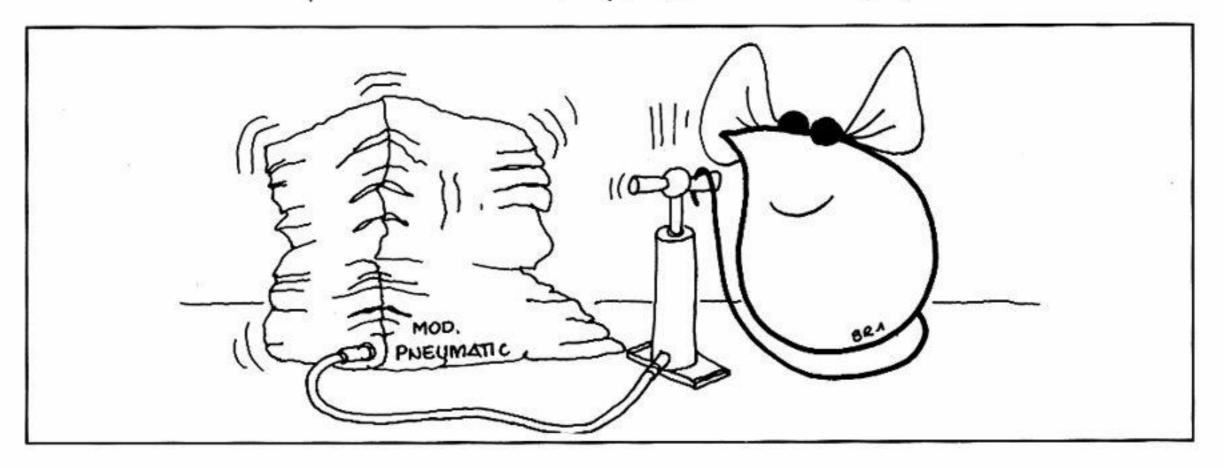
Diventare programmatori

Ho notato che recensite molti libri per imparare a programmare Amiga.

Esistono volumi validi anche per il C/64?
(Emanuele Giacometti - S.M. di Sala)

Ti posso consigliare la **Guida di riferimento per il programmatore** del C/64, edito dalla stessa Commodore, in lingua italiana.

Purtroppo il volume non è facilmente rintracciabile nelle librerie.

In alternativa ti posso suggerire di procurarti gli arretrati che mancano alla tua collezione di C.C.C. e che rappresentano una vera e propria miniera di trucchi, sia in Basic che in linguaggio macchina.

Se, però, hai deciso di imparare seriamente a programmare, ti consiglio di valutare con attenzione la possibilità di operare su computer Ms-Dos oppure Amiga; credimi, è molto più affascinante.

SYSTEMS EDITORIALE PER TE

La voce
Aggiunge al C/64 nuovi comandi Basic che consentono sia di far parlare il computer, sia di farlo Cantare! Diversi esempi allegati.
Cassetta: L. 12000 - Disco: L. 15000

Raffaello
Un programma completo per disegnare, a colori, con il C/64: linee, cerchi, quadrati, eccetera. Valido sia per disegno a mano libera che geometrico.
Cassetta: L. 10000

Oroscopo
Devi solo digitare la data di nascita e le coordinate geografiche del luogo che ti ha dato i natali. Vengono quindi elaborate le varie informazioni (case, influenze dei segni astrali, eccetera) e visualizzato un profilo del tuo carattere. Valido per qualsiasi anno, è indicato sia agli esperti sia ai meno introdotti. E' allegata una tabella delle coordinate delle più note città italiane e l'elenco delle ore legali in Italia dal 1916 al 1978.
Cassetta: L. 12000 - Disco: L. 12000

Computer Music
Cassetta contenente numerosi brani di successo da far eseguire, in interrupt, al tuo C/64 sfruttando, fino in fondo, il suo generatore sonoro (SID).
Cassetta: L. 12000

Gestione Familiare
Il più noto ed economico programma per controllare le spese e i guadagni di una famiglia.
Cassetta: L. 10000 - Disco: L. 10000

Banca Dati
Il più noto ed economico programma per gestire dati di qualsiasi natura.
Cassetta: L. 10000 - Disco: L. 10000

Matematica finanziaria
Un programma completo per la soluzione dei più frequenti problemi del settore.
Cassetta: L. 10000 - Disco: L. 20000

Analisi di bilancio
Uno strumento efficace per determinare con precisione i calcoli necessari ad un corretto bilancio.
Cassetta: L. 10000 - Disco: L. 20000

Corso di Basic
Confezione contenente quattro cassette per imparare velocemente le caratteristiche delle istruzioni Basic del C/64 e i rudimenti di programmazione. Interattivo.
Cassetta: L. 19000

Corso di Assembler
Un corso completo su cassetta per chi ha deciso di abbandonare il Basic del C/64 per addentrarsi nello studio delle potenzialità del microprocessore 6502. Interattivo.
Cassetta: L. 10000

Logo Systems
Il linguaggio più facile ed intuitivo esistente nel campo dell'informatica; ideale per far avvicinare i bambini al calcolatore.
Diversi esempi allegati.
Cassetta: L. 6500

Compilatore Grafico Matematico
Uno straordinario programma compilatore, di uso semplicissimo, che permette di tracciare, sul C/64, grafici matematici Hi-Res ad altissima velocità. Esempi d'uso allegati.
Cassetta: L. 8000

Emulatore Ms-Dos e Gw-Basic
Un prodotto, unico nel suo genere, che permette di usare, sul C/64 dotato di drive, la sintassi tipica del più diffuso sistema operativo del mondo. Ideale per studenti.
Solo su disco: L. 20000

Emulatore Turbo Pascal 64
Permette di usare le più importanti forme sintattiche del linguaggio Turbo Pascal (anche grafiche!) usando un semplice C/64 dotato di drive.
Ideale per studenti.
Disco: L. 19000

Speciale drive
Questo speciale fascicolo costituisce una guida di riferimento per le unità a disco del C64/128.
Comprende anche un velocissimo turbo-disk più la mappa completa della memoria del drive.
Fascicolo + disco: L. 12000

Utility 1
Un dischetto pieno zeppo di programmi speciali per chi opera frequentemente con il drive.
Disco: L. 12000

Utility 2
Seconda raccolta di utility indispensabili per realizzare sofisticate procedure di programmazione.
Disco: L. 15000

Graphic Expander 128
Per usare il C/128 (in modo 128 e su 80 colonne) in modo grafico Hi-res. Aggiunge nuove, potenti istruzioni Basic per disegnare in Hi-Res con la massima velocità in modalità 80 colonne.
Disco: L. 27000

Directory
Come è noto, a partire dal N. 10 di "Software Club" (la rivista su disco per l'utente dei "piccoli" computer Commodore), vengono riportati tutti i listati, in formato C/64-C/128, pubblicati su "Commodore Computer Club".
In precedenza tali listati venivano inseriti, mensilmente, in un dischetto, di nome "Directory", che oltre ai programmi di C.C.C. ospitava decine di altri file tra cui musiche nell'interrupt, giochi, listati inviati dai lettori e altro.
Ogni disco, dal prezzo irrisorio, contiene quindi una vera miniera di software. Ordinando i dischetti di "Directory" si tenga conto che al N. 1 corrispondeva il contenuto del N. 34 di "Commodore Computer Club", al N. 2 il N. 35 e così via.
Ogni dischetto: L. 10000

Super Tot '64
La nuova e completa edizione del programma Tot 13 con tutti i sistemi di riduzione e di condizionamento.
Ampia sezione dedicata alla teoria.
fascicolo + disco: L. 15000

Amiga Totospeed
Finalmente anche per Amiga un programma orientato alla compilazione delle schedine totocalcio.
Fai tredici conil tuo Amiga.
disco: L. 20000

SYSTEMS EDITORIALE PER TE

Disk'o'teca
Grazie a questa nutrita raccolta di brani musicali potrete divertirvi ascoltando i migliori brani prodotti dai vostri beniamini, oltre a una serie di composizioni prodotte "in casa".
In omaggio un bellissimo poster di Sting.
Disco: L.15.000

Assaggio di primavera
Esclusivo!
In un'unica confezione potrete trovare ben due cassette di videogiochi assieme a un comodo e funzionale joystick.
Cassette: L. 15.000

LIBRI TASCABILI

64 programmi per il C/64
Raccolta di programmi (giochi e utilità) semplici da digitare e da usare. Ideale per i principianti. **(126 pag.)**
L. 4800

I miei amici C/16 e Plus/4
Il volumetto, di facile apprendimento, rappresenta un vero e proprio mini-corso di Basic per i due computer Commodore. Numerosi programmi, di immediata digitazione, completano la parte teorica. **(127 pag.)**
L. 7000

62 programmi per C/16, Plus/4
Raccolta di numerosi programmi, molto brevi e semplici da digitare, per conoscere più a fondo il proprio elaboratore.
Ideale per i principianti. **(127 pag.)**
L. 6500

Micro Pascal 64
Descrizione accurata della sintassi usata dal linguaggio Pascal "classico". Completa il volume un programma di emulazione del PL/O sia in formato Microsoft sia in versione C/64 (da chiedere, a parte, su disco). **(125 pag.)**
L. 7000

Dal registratore al Drive
Esame accurato delle istruzioni relative alle due più popolari periferiche del C/64. Diversi programmi applicativi ed esempi d'uso. (94 pag.)
L. 7000

Il linguaggio Pascal
Esame approfondito della sintassi usata nel famoso compilatore. (112 pag.)
L. 5000

Simulazioni e test per la didattica
Raccolta di numerosi programmi che approfondiscono e tendono a completare la trattazione già affrontata sul precedente volume. **(127 pag.)**
L. 7000

Dizionario dell'Informatica
Dizionario inglese-italiano di tutti i termini usati nell'informatica. (Edizione completa). **(385 pag.)**
L. 10000

Word processing: istruzioni per l'uso
Raccolta delle principali istruzioni dei più diffusi programmi di w/p per i sistemi Ms-Dos: Word-Star, Samna, Multimate Advantage, Word 3. **(79 pag.)**
L. 5000

Unix
Un volumetto per saperne di più sul sistema operativo professionale per eccellenza.
Un necessario compendio per l'utente sia avanzato che inesperto (**91 pag.)**
L. 5000

ABBONAMENTO

Commodore Computer Club
11 fascicoli: L. 50.000

ARRETRATI

Ciascun numero arretrato di C.C.C. L. 6.000

Come richiedere i prodotti Systems

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, L. 3500 per spese di imballo e spedizione, oppure L. 6000 se si desidera la spedizione per mezzo raccomandata.
Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L. 50000.
Per gli ordini, compilare un normale modulo di C/C postale indirizzato a:

C/C Postale N. 37 95 22 07
Systems Editoriale Srl
Via Mosè, 22
20090 Opera (MI)

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento"), non solo il vostro nominativo completo di recapito telefonico, ma anche i prodotti desiderati ed il tipo di spedizione da effettuare.
Per sveltire la procedura di spedizione sarebbe opportuno inviare, a parte, una lettera riassuntiva dell'ordine effettuato, allegando una fotocopia della ricevuta del versamento.
Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare, oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a:

Systems Editoriale
Milano

IL PASSATO E' PRESENTE

L'affannosa corsa all'illusoria conquista di esclusive fette di mercato fa spesso dimenticare particolari di notevole importanza

di Alessandro de Simone

Poche settimane fa, un autorevole periodico di informatica rendeva pubbliche le innocenti ammissioni dei progettisti di un nuovo microprocessore.

Sembra, anzi è certo, che in alcuni casi, anzi frequenti, il microprocessore in questione si inchioda, senza apparente motivo; e con esso, ovviamente, l'intero sistema da esso supportato.

La notizia non farebbe alzare il sopracciglio più di tanto nemmeno al più smaliziato esperto informatico; di un nuovo chip si sa come è fatto, ma non sempre come funziona; per soddisfare quest'ultima curiosità, assicurano, si stanno attrezzando.

Il guaio è che non si tratta di un microprocessore in fase sperimentale, ma di una "macchina" già venduta, distribuita e montata su computer che trovano posto sulle scrivanie di ignari utenti finali.

La fabbrica di processori, anzi, è stata costretta ad ammettere la presenza del **bug** proprio a causa dell'inspiegabile inchiodamento di tutti gli elaboratori posti in commercio. Inoltre ha anche candidamente ammesso che altri bug, non rilevati dagli utenti, si verificavano nelle precedenti versioni di chip (anch'essi attualmente "funzionanti" su macchine già vendute), ma sono stati eliminati nella versione più recente; chi lo desidera, ovviamente, può chiedere la sostituzione del chip(!).

Profitto ad ogni costo

Viene spontaneo chiedersi come mai una così prestigiosa casa costruttrice (il cui nome taciamo per pietà) non abbia effettuato i necessari esperimenti prima di mettere in commercio il chip incriminato. Ci si può chiedere, giacchè ci siamo, se e quali altri bug possono presentarsi, ed in quali occasioni.

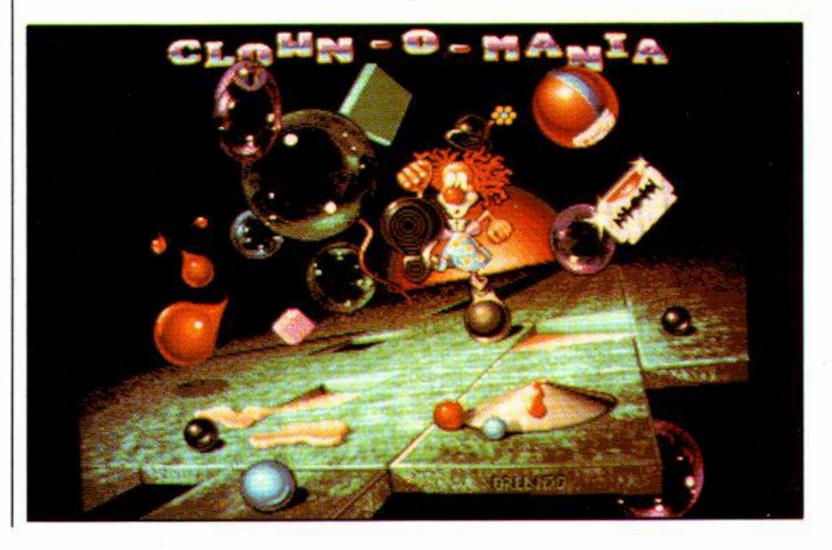
Il tormentato mercato informatico vive, più di altri, grazie ad una frenesia globale difficilmente immaginabile in ambienti economici tradizionalmente più tranquilli, come quelli dei ravanelli o del sale da cucina. I giapponesi, infatti, premono alle porte; i taiwanesi attaccano dai fianchi e la Silicon Valley vede diminuire sempre più l'efficacia delle difese tecnologiche, in primo luogo, ed economiche, soprattutto. Un esempio ne è la guerra delle Ram, che costringe milioni di utenti a rinviare l'acquisto di espansioni di memoria a causa del loro elevato **prezzo** (e non del loro **costo**, badate bene).

L'unico modo di accaparrarsi il mercato, è ormai quello di crearlo apposta per diventarne esclusivi padroni.

Nessun utente finale (rabdomante, ragioniere, studente o semplice appassionato che sia) sente il bisogno di buttare alle ortiche il suo Ms-Dos basato sull'80286; si accontenterebbe di spremerlo un altro po'.

Nessun utente di Amiga 1000, d'altra parte, avrebbe sentito il desiderio di "passare" al 500 o al 2000 se il firmware del 1000 fosse stato più affidabile ed il software di base più completo e realmente compatibile con gli sviluppi futuri.

Sembra che solo i C/64isti non hanno più diritto a chiedere di più: tutto ciò che si poteva spremere da un 8 bit è stato spremuto e, quando non ce la faceva da solo, sono state realizzate cartucce, velocizzatori ed accessori vari di più o meno fortunata evoluzione.

Nessuno, in conclusione, sentiva la necessità di un sistema basato su di un processore più veloce di un 80386 (da considerarsi ancora "sperimentale", sotto molti punti di vista); eppure hanno pensato bene di accontentare questi mancati desideri, di realizzare una macchina specifica e di lanciare il nuovo standard.

Topi e tastiere

L'inverosimile brama di denaro ha già fatto le sue vittime: tra queste citiamo, soprattutto, gli utenti finali che, per questioni prettamente commerciali, sono inconsapevolmente costretti ad acquistare software sempre più potenti (a prezzo corrispondente) che utilizzeranno tutt'al più al 20% delle loro effettive potenzialità.

Volendo citare le schede grafiche per PC, poi, i cadaveri illustri si sprecano. Oggi l'utente finale, che desidera entrare in possesso di un sistema Ms-Dos, si perde nella marea di schede grafiche e monitors di caratteristiche rigorosamente incompatibili tra loro. Va a finire, molto spesso, che ci si porta a casa la configurazione - base (monocromatica) in attesa di saperne di più ed effettuare, giustamente, un acquisto oculato.

Tanto per restare in argomento, sarebbe bene che l'utente finale, e soprattutto chi ha avuto la pazienza di leggere fino a questo punto, si rendesse conto delle varie curiosità riscontrabili nel settore informatico.

Prendiamo, ad esempio, la famosa rivoluzione portata dal **mouse** e dalle **icone**. Chi ha lavorato con Amiga, o con un qualsiasi sistema basato sulle intuitive immaginette presenti su video, sa benissimo che la gestione delle risorse del computer è di gran lunga più intuitiva rispetto al tradizionale sistema di input (digitazione del comando e pressione del tasto Return).

Tale comodità si traduce immediatamente in una maggior diffusione dell'informatica tra fasce di popolazione che, prima, diffidavano del mezzo informatico a causa della presunta difficoltà di "colloquio". Le conseguenze economiche di tale diffusione non dovrebbero sfuggire a nessuno. Pur di contrastare il **McIntosh**, tuttavia, fino a poco tempo fa, autorevoli esponenti di autorevoli case costruttrici affermavano, senza apparente vergogna: **"Real men don't use Icons"** (I veri uomini non usano le icone) dimostrando di voler legare la tastiera, freudianamente, ad organi umani digitalizzabili con estrema difficoltà.

Subito dopo (per la cronaca) la stessa autorevole casa costruttrice basò le architetture dei suoi elaboratori sull'indissolubile binomio mouse - icone.

Ma permettetemi di continuare a parlare di schermi.

Una finestra, anzi tutte

Tutti noi siamo rimasti affascinati dalla possibilità, offerta da Amiga, di creare finestre di varie dimensioni e di posizionarle ovunque sullo schermo.

Pochi di noi, però, si sono soffermati sulla reale necessità di creare un numero pressochè infinito di finestre (ad esempio, come individuarne una tra le altre, infinite?); non ci sarebbe certo venuto un collasso se le finestre potevano avere "solo" una diecina di dimensioni prefissate; avremmo continuato a vivere tranquilli anche se le finestre potevano essere posizionate "solo" in una dozzina di posizioni prestabilite.

La limitazione delle features avrebbe consentito (forse) una più efficiente e veloce gestione delle routines grafiche di base.

Di numero ancora minore sono gli utenti che si rendono conto della estrema lentezza di gestione delle finestre, quando queste risultano in numero eccessivo.

Provate, infatti, a "lanciare" una dozzina di **Clock** (il divertente orologio presente nel cassetto Utilities del Workbench), visualizzateli in modo che siano parzialmente sovrapposti tra loro e provate a modificare la dimensione di uno solo di essi: quanto tempo passa prima che lo schermo si ricomponga?

Morale della favola: inutile offrire potenzialità che non possono, in pratica, essere utilizzate con profitto.

Al contrario, cerchiamo di capire perchè la "moda" delle finestre ha preso piede.

Storia del monitor

La quantità di informazioni gestibili con i moderni pacchetti software trova una grossa limitazione nelle anguste 80 colonne e 25 righe dei nostri monitors.

Come fare, si sono chiesti i progettisti di software, per far apparire (ad esempio) un menu senza far scomparire la videata presente sullo schermo?

Semplice: si sovrappone una porzione di schermo (finestra) contenente il menu; effettuata la scelta, la finestra (**window**) scompare facendo riapparire quanto c'era in precedenza, testo o grafico che fosse.

La faccenda, però, ha preso la mano ai pregettisti ed il numero di finestre apribili, in casi particolari, è talmente elevato che la videata iniziale scompare del tutto alla vista dell'utente.

A volte, se non bastasse, il fattore estetico prende il sopravvento sulla razionalità. Lavorando con il DTP Ventura, ad esempio, appaiono finestre enormi, perfettamente definite ma estremamente coprenti; una cornice in meno, una sub- window in più ed ecco che la preziosa area di schermo verrebbe coperta solo quanto basta. E meno male che non hanno pensato a disegnare anche le tapparelle!

Ma, a pensarci bene, perchè mai ricorrere alle finestre?

La risposta è stata data all'inizio: perchè i monitors sono piccoli.

Ma, allora, perchè non realizzare monitors specifici per computers?

E' probabile che il lettore abbia voluto rileggere un paio di volte la frase precedente: *tutti i monitors* - si sarà detto - *sono specifici per i nostri computers!*

Pochi riflettono, invece, sul fatto che i monitors attuali non sono altro che i diretti discendenti dei televisori in bianco e nero degli anni '40.

Quando inventarono la televisione, infatti, si pose subito il problema delle dimensioni da assegnare allo schermo, in modo da creare uno standard di riferimento.

La scelta cadde sul rapporto base / altezza delle pellicole cinematografiche usate in quei tempi lontani. In questo modo, infatti, era possibile trasmettere, senza alcuna difficoltà, films e roportages effettuati con macchine da ripresa che costituivano già uno standard consolidato.

Se avessero inventato la TV quando esplose il Cinemascope, i nostri schermi sarebbero più stretti e lunghi; ce ne accorgiamo, comunque, quando trasmettono film più moderni.

Al momento dell'invenzione del Cinemascope, purtroppo(?), la frittata era già fatta. Nessun costruttore si assunse il rischio di porre in vendita apparecchi televisivi (e telecamere ed elettronica di supporto varia) destinati ad un probabile insuccesso commerciale.

Ne seguì un ulteriore consolidamento del formato, che è rimasto tal quale fino ai giorni nostri.

La stessa sorte è toccata all'informatica. Quando il progresso elettronico lo consentì, fu scelto, come sistema di visualizzazione, ancora quello televisivo.

Furono così realizzate schede elettroniche che, collegando elaboratore a schermi video, permettevano di abbandonare le telescriventi a favore di un più rapido, efficiente ed economico sistema di visualizzazione.

Le schede elettroniche erano già abbastanza dispendiose per affrontare la progettazione e la costruzione di standard più efficienti di quelli TV, e così il formato televisivo divenne realmente immortale.

Ma, ci si può chiedere, quale dimensione era preferibile alla TV? Ma quella di un foglio di carta, diamine!

Spesso diementichiamo che il computer, usato professionalmente, dovrebbe tendere a sostituire una comune scrivania; quest'ultima, guarda caso, è piena di fogli di dimensioni standard, tipicamente A4 (cm. 21 X 29.6). Usando parole meno tecniche, chiunque sa che un foglio dattiloscritto contiene, tradizionalmente, 60 righe di 80 caratteri ciascuna. Uno schermo video, invece, ha il formato di 80 X 25.

Se, dunque, hanno adeguato la TV ai film in bianco e nero (a quei tempi in voga) perchè non hanno fatto lo stesso per i monitor dei computer?

La risposta è pleonastica: a quei tempi la tecnologia non lo consentiva, se non a costi proibitivi.

Rimane il fatto che oggi uno schermo di formato A4, realmente professionale, è un fuori-standard e, come tale, costa un mucchio di soldi; per non parlar del fatto che i software in grado di gestirli sono pochi; e cari anch'essi.

Un monitor, anzi due

Riassumendo, i sistemi di visualizzazione attuali costringono ad usare il computer in modo poco professionale, dal momento che non riescono a "simulare" efficacemente il ***media*** tradizionale di ogni ufficio: il foglio di carta.

E' probabile, anzi auspicabile, che la diffusione ulteriore dell'informatica tenda a far ridurre il consumo di carta e a far dimenticare il formato di questo storico media. Fino al 2000, però, dubito che si possa fare a meno delle comunicazioni scritte.

La soluzione al problema della presenza contemporanea di numerose informazioni sembra, quindi, essere irrimediabilmente legata al trucco delle finestre. Nessuno ha pensato, però, che ormai i monitors costano piuttosto poco e che la presenza di due monitor affiancati potrebbe risolvere, almeno in parte, il problema.

In effetti numerose software house, specializzate soprattutto in grafica, hanno scelto tale soluzione proponendo stazioni di lavoro dotate di ben tre schermi.

Il principale, di notevoli dimensioni, è quello destinato all'area di lavoro; gli altri due, di dimensioni minori, contengono ulteriori informazioni più o meno utili.

Si potrebbe obiettare che una tale scelta (da parte di software house che

sviluppano programmi per posti stand-alone) potrebbe rivelarsi un vero insuccesso dal momento che nessuna casa costruttrice propone sistemi dotati di due monitors. Bisognerebbe, per avere qualche speranza di vendita, che venisse "spinto" l'acquisto di nuovi elaboratori in grado di gestire più monitors.

Ebbene, il ragionamento non è totalmente valido.

Per spiegarci meglio, consideriamo un tipo di pacchetto piuttosto diffuso, come può essere un Word Processor oppure un D.T.P. Al momento della sua installazione potrebbe esser posta una semplice domanda, del tipo...

"Quale sistema di visualizzazione preferisci?

1- Finestre (schermo singolo)

2- Monitor supplementare

Il programma, ovviamente, terrebbe conto della risposta data alla domanda. Se si è preferito il sistema a finestre, le informazioni verranno scaricate su una porzione di schermo (window) nè più nè meno di come avviene attualmente.

Nel caso, invece, si sia scelto il monitor supplementare, le stesse informazioni verrebbero "dirottate" verso una porta seriale.

E non ditemi che la programmazione di un simile switch software costa troppa fatica! Allo stesso modo non possiamo credere che il costo di una interfaccia seriale (circa L. 40000, al dettaglio!) farebbe aumentare eccessivamente il prezzo di un computer dotato di tali caratteristiche.

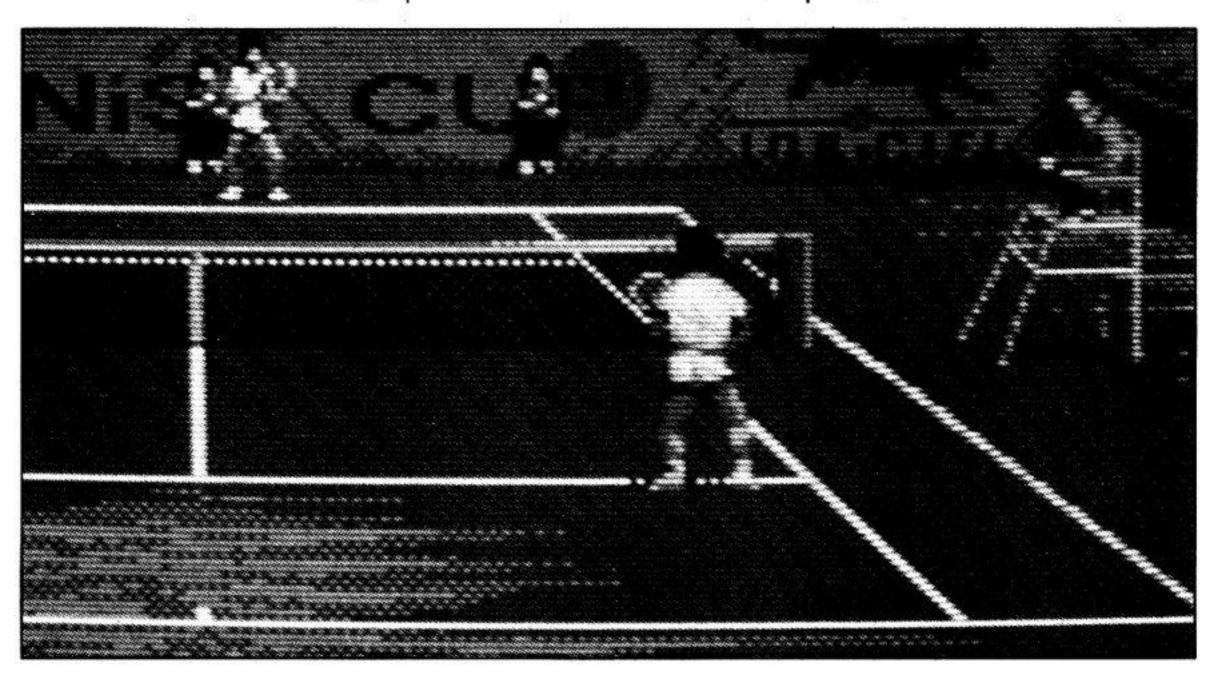
Il monitor supplementare, ovviamente, dovrebbe possedere una scheda elettronica in grado di interpretare i segnali in arrivo da una Rs-232; ma una Rom contenente il firmware ed i set di caratteri, qualche Kram per mappare lo schermo, un microprocessore (un 8 bit è più che sufficiente) ed uno sprite per seguire il mouse ed il gioco è fatto; si potrebbe, ovviamente, pensare al futuro e realizzare uno standard ben fatto, in grado di gestire anche grafica e, perchè no?, colore.

Con la tecnologia attuale, insomma, si potrebbe offrire, senza elevati rischi di insuccesso commerciale, pacchetti software in grado di gestire due schermi, senza alterare di una sola virgola l'architettura di base di un qualsiasi computer. Molti utenti, pertanto, sarebbero felicissimi di acquistare un secondo monitor, magari solo in bianco e nero, su cui "scaricare" le indispensabili informazioni finora relegate nelle finestre; il monitor principale, invece, sarebbe destinato esclusivamente a contenere il testo, la tabella o la "maschera" su cui si lavorerebbe con maggiore libertà.

Già oggi, del resto, troviamo pacchetti in grado di gestire contemporaneamente due stampanti (cioè due porte parallele) e/o due porte seriali (cui sono collegati, ad esempio, mouse e modem).

Ciò significa, quindi, che è già maturata un'esperienza tale da consentire l'invio selettivo di gruppi di informazioni.

Conclusioni

Troppo spesso le caratteristiche tecnologiche di alcune apparecchiature fanno gridare al miracolo; pochi, però, si rendono conto che i sofisticati artifici software non sono altro che palliativi idonei a nascondere errori commessi in precedenza, e non più rimediabili. Il guaio è che ancora oggi, nonostante l'esperienza acquisita, si continua sulla stessa strada pensando che, in futuro, qualcuno inventerà qualcosa per sopperire alle attuali carenze. Sarebbe la stessa cosa, ad esempio, eliminare i segnali di Stop ed i semafori; e consigliare alla gente di guidare con il casco aumentando, nel contempo, le tariffe dell'assicurazione obbligatoria...

NONSOLOMODEM

Una panoramica sui risvolti pratici di uno degli aspetti più affascinanti dell'informatica: la comunicazione

di Carlo d'Ippolito

Viviamo nell'era del *villaggio globale* e le grandi distanze non costituiscono più un ostacolo insuperabile.

Le barriere linguistiche sono state abbattute ricorrendo ad un linguaggio universale (l'inglese), oppure a traduzioni simultanee, ove occorra.

La tecnologia applicata all'elettronica consente poi di trasferire informazioni, sotto forma di *voce, testi* o *immagini*, su lunghe distanze, ed in tempi molto più brevi rispetto a quelli della normale comunicazione su carta (lettere o giornali).

Telefono, telex, telefax e *reti radiotelevisive* sono usati quotidianamente per il trasferimento delle informazioni, da una parte all'altra del globo.

Queste tecniche appartengono alla categoria delle "telecomunicazioni".

TI MANDO UN FAX

Il computer si serve di particolari strumenti per comunicare con l'uomo: il monitor innanzitutto, poi le stampanti, per restituire i dati scritti "nero su bianco", in definitiva la collocazione delle parole a noi più familiare.

Per facilitare ulteriormente il dialogo con il suo utilizzatore, il computer, con lo scanner, ha *imparato* a leggere.

Due sono gli impieghi del computer: l'uno, come elaboratore locale, "assoluto", nel senso latino di indipendente, libero da legami; l'altro come terminale intelligente per reti di dialogo esterno più o meno ampie. Proprio quest'ultimo, per l'informatizzzione dei principali ambienti di lavoro, per tutti quello commerciale, ha reso indispensabile un sistema internazionale di comunicazione fra computer, il **data communication**.

Laddove lo scambio di informazioni elaborate è vitale per il positivo andamento degli affari, si è diffusa l'**office automation**, vale a dire un sistema che integra telecomunicazioni, trasmissioni di dati, attrezzature da ufficio e lui, immancabile, insostituibile, il computer.

Uno dei più semplici esempi di collegamento tra unità diverse, è quello relativo al "colloquio" tra un computer ed una stampante.

Occorre un cavo di media lunghezza, e, qualunque sia l'interfaccia di output usata, seriale o parallela, i dati saranno trasferiti dalla memoria del computer alla stampante alla velocità consentita dalle caratteristiche del suo buffer e della velocità di stampa.

La trasmissione di dati fra due computer si ottiene, allo stesso modo, connettendo le interfacce input e output, generalmente Rs-232.

Ma se le esigenze di scambio dati vanno oltre le distanze consentite dai pochi metri di cavo, come si può fare?

Può capitare, infatti, di dover colloquiare con un computer posto in altro ufficio, se non in un'altra città.

Soluzione di questo problema potrebbe essere quella di copiare i dati su floppy disk e poi spedirlo per posta a chi ne ha bisogno.

Tale strattagemma richiede, ovviamente, tempi brevi e servizi postali affidabili; ne consegue che, in Italia, è consigliabile ricorrere a un canale di comunicazione di dati.

Anche questo collegamento, detto *remoto*, avviene attraverso interfacce seriali.

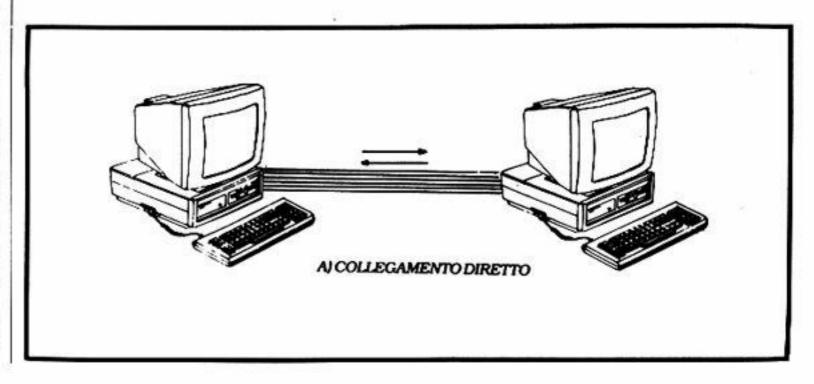
Esiste una analogia tra il modo di comunicare proprio degli uomini e quello dei computer: due persone possono infatti comunicare faccia a faccia, scambiandosi lettere o usando il telefono, che è una rete di comunicazione SIP permettendo.

La scelta fra le tre alternative è determinata da fattori come distanza, urgenza, praticità, affidabilità e costo.

La trasmissione di dati è una delle più innovative e interessanti caratteristiche dei personal computer. Come il telefono, consente di accedere al cosiddetto *mondo esterno* senza spostarsi dal proprio posto di lavoro.

Qualsiasi trasmissione di dati richiede un "protocollo", una procedura di comportamento da seguire per accedere a un determinato sistema di lavoro.

Può essere utile un esempio: per spedire una lettera sono indispensabili il

A) COLLEGAMENTO DIRETTO

francobollo, l'indirizzo del destinatario, uno stabilito formato della busta.

Questo è un protocollo.

Altro esempio, è il **Midi**, un linguaggio interpretato da tutti gli strumenti musicali che consente, per esempio, il collegamento tra una tastiera e un computer.

IL MODEM

Prima dell'avvento degli elaboratori, l'elemento comune agli uffici di tutto il mondo era il telefono.

Proprio grazie alle reti telefoniche si attua il collegamento via modem.

Studiamone il funzionamento: la voce umana genera un segnale a frequenza variabile, detto *analogico*, e infatti le linee telefoniche, concepite per la comunicazione vocale, trasmettono solo dati analogici.

Il modem è un apparecchio in grado di trasformare, o **mod**ulare, i segnali digitali emessi dai computer che, per essere interpretati da una linea telefonica, devono essere trasformati in analogici. Alla fine del collegamento il procedimento è inverso: i dati analogici sono ritrasformati, **dem**odulati, in digitali.

Il nome modem deriva proprio dalla funzione di modulazione - demodulazione. Oltre al costo del modem, si paga solo il tempo per cui viene utilizzata la linea telefonica.

Quasi tutti i modem sono apparecchi *separati* dall'elaboratore, collegati all'interfaccia seriale di output. Alcuni sono invece a *scheda*, occupano uno slot di espansione e si interfacciano direttamente con il bus interno del computer. Sono meno costosi dell'equivalente unità separata.

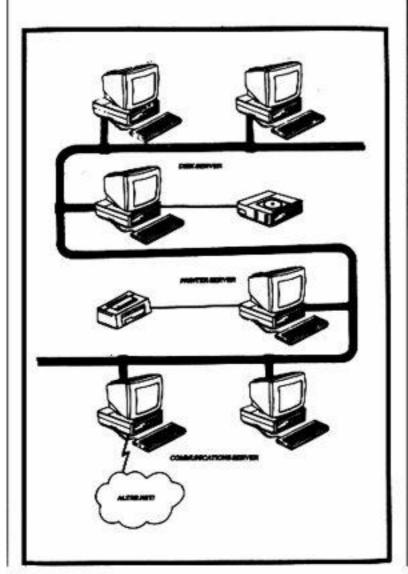
L'unità di misura della velocità di modulazione si esprime in **baud**: coincide con il numero di bit al secondo che passano attraverso la connessione seriale. I tipi di trasmissione sono due: **sincrona** e **asincrona**.

Il primo prevede l'invio dei caratteri sotto forma di sequenze continue di bit; alcuni caratteri speciali, detti sincronizzatori, informano l'unità ricevente in merito all'inizio e alla fine di ogni sequenza. Nella seconda modalità, la più usata da personal computer e da terminali a velocità medio-bassa, ogni singolo carattere è preceduto, e seguito, da un bit di partenza e da uno di arrivo.

Una valida alternativa al modem, anche se non molto diffusa, è costituita dagli accoppiatori acustici, che convertono i segnali digitali direttamente in acustici e viceversa, rendendo così possibile l'utilizzo delle reti telefoniche.

DATO CHE VA, DATO CHE VIENE

I grandi mainframe e le reti informatiche hanno potenti capacità di elaborazione e conservano enormi quantità di informazioni. Per usufruirne servono i terminali, unità programmabili dotate di periferiche e dispositivi locali di memorizzazione. Consentono di combinare la potenza locale di calcolo di un personal computer con l'accesso diretto a database remoti di grandi sistemi, con la possibilità di comunicare con altri utenti dello stesso sistema. Perchè ciò avvenga occorre che le macchine collegate siano integrate con i protocolli della rete, cioè che siano uniformate alle regole della rete stessa. Il processo di emulazione conferisce al terminale che si vuole integrare, tramite convertitori di protocollo hardware e software, la capacità di colloquio con reti a largo raggio private e pubbliche.

I **public data network** rappresentano l'equivalente digitale delle linee telefoniche; trasmettono agli abbonati informazioni digitali anzichè analogiche (a voce). **Videotex** e **teletex**, ad esempio, offrono l'accesso a banche dati di sistemi commerciali, dotate di alte velocità di comunicazione ed elevati livelli qualitativi: informazioni finanziarie, di attualità, sport, viaggi e cultura, sono gli argomenti maggiormente trattati.

NEL NOSTRO PICCOLO

Il movimento di informazioni maggiore avviene in ambito limitato, in ambienti ristretti che costituiscono le **reti locali**. Ne sono un esempio chiaro i molti uffici di un'azienda situati su diversi piani dello stesso edificio o di costruzioni circostanti. I **Private Branch Exchange** sono i centralini privati telefonici, ai quali corrispondono, nel settore informatico le **Lan**, **Local Area Network**, che trasmettono dati digitali, e quindi non usano modem, mettendo in comune risorse hardware come stampanti laser e grandi memorie su disco fisso. La struttura di una Lan può essere ad **anello**, a **stella**, o a **bus**.

Quest'ultima è la più frequente: i dati viaggiano a velocità comprese tra 0,5 e 10 milioni di bit al secondo, maggiore rispetto ai valori delle linee telefoniche citate prima, da considerare le Lan come estensioni logiche del bus interno. Infatti i programmi applicativi che girano su uno dei terminali della rete, non incontrano alcuna difficoltà nell'accedere alle memorie di massa o alle periferiche del sistema.

Il dispositivo che permette il collegamento alla rete di una o più risorse in comune si chiama **Server**. La funzione di server può essere assolta da uno dei personal della rete o da un'apparecchiatura dedicata.

Qualunque sia la loro configurazione fisica, i server svolgono una delle seguenti mansioni logiche: file server, controllore dell'accesso alle memorie di massa e relativo impiego; printer server, inerente alla gestione delle stampanti; communication server, ovvero convertitore dei protocolli di comunicazione, quando è necessario, al fine di regolare i canali di trasmissione.

Per un utilizzo ottimale delle potenzialità di una rete, i software di sistema e applicativi richiedono adattamenti alle condizioni di lavoro in cui si trovano ad operare.

CAMPUS

64 / 128

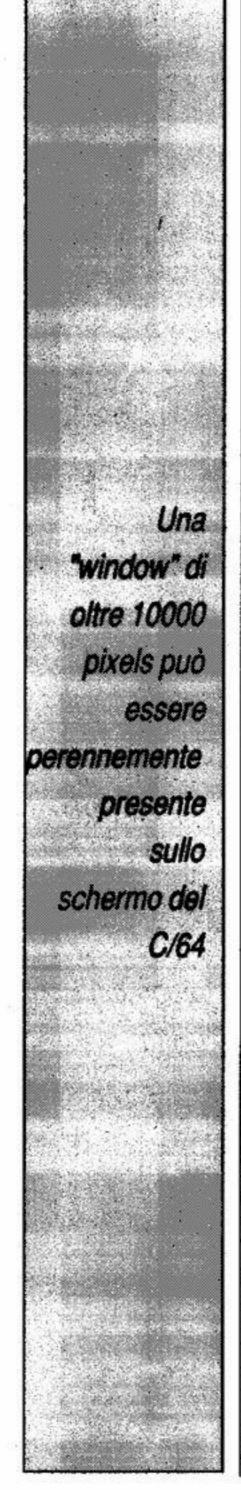
18 - COME TI GESTISCO UN BLOCCO DI SPRITES

Le sprites sul C64 sono sempre visualizzabili sia in modo testo che in grafica. Inoltre il circuito VIC-II è in grado di gestirne otto contemporaneamente sullo schermo.
Il programma proposto consente di posizionare una sorta di "finestra" grafica in un punto qualunque dello schermo formata dalle otto sprite, appunto.
Le routine, interamente scritte in linguaggio macchina, consentono di implementare efficaci procedure di plot sulla mini pagina grafica visualizzata.

26 - UNA ZEBRA A COLORI

Potevamo stupirvi con effetti spettacolari e colori ultravivaci... e l'abbiamo fatto!
Particolarmente dedicato agli esperti, che possono maneggiarlo in modo completo, questo programma consente di visualizzare divertenti effetti speciali sul monitor del vostro C64.
I lettori "novellini" potranno divertirsi digitando comunque il listato (in assembler) e osservando gli effetti risultanti.

30 - UN MESSAGGIO LAMPEGGIANTE

Particolarmente adatto alle presentazioni, questo programma consente di visualizzare nelle prime quattro righe dello schermo un messaggio caratterizzato da una variazione graduale del colore.
Non è richiesto un elevato livello di esperienza per capire questo programma che quindi si rivela un ottimo banco di prova sia per l'utente esperto che per il "novellino" alle prese con l'assembler.

COME TI GESTISCO UN BLOCCO DI SPRITES

Una fra le caratteristiche peculiari del circuito VIC-II è la possibilità di gestire contemporaneamente fino ad otto animazioni. Le routines in linguaggio macchina presentate sono finalizzate alla gestione complessiva degli sprites, come se si trattasse di un'unica piccola pagina in alta risoluzione

di Tito Brasolin

Le routines in linguaggio macchina di queste pagine gestiscono gli otto sprites del C/64 come se si trattasse di un'unica piccola pagina in alta risoluzione, che possiede il vantaggio di essere visualizzata dovunque sullo schermo, in qualunque modo grafico ci si trovi.

Le due routines, inoltre, sono perfettamente rilocabili.

La prima delle due è in grado di riportare una stringa (eventualmente di dimensioni espanse) all'interno del blocco di sprites.

Ovviamente essa comprende vari controlli e conteggi, ma il problema posto alla sua base è il **plottaggio** di un punto.

La procedura incaricata di plottare un punto all'interno di un gruppo di sprites deve tenere conto del fatto che essi possono essere variamente disposti.

La teoria

In generale noi considereremo un certo numero (**NUM**) di sprites, disposti a righe di **XL** sprites ciascuna, e numerati progressivamente da sinistra a destra e dall'alto verso il basso.

Sia dunque **XL** il numero di sprites per riga, **(X, Y)** le coordinate del punto riferite all'angolo superiore sinistro. Per prima cosa occorre sapere in quale riga si trova lo sprite contenente il punto considerato. Questo valore è fornito dal risultato intero della divisione Y / 21. Cioè...

RIGA = int (Y / 21)

In secondo luogo occorre sapere in quale sprite della riga calcolata si trova il punto. Tale sprite sarà...

SPRITE = int (X / 24)

L'ordinata del punto relativa all'angolo superiore sinistro dello sprite in cui esso è contenuto è data dal resto della prima divisione, cioè...

YR = Y - 21 * INT (Y / 21)

L'ascissa relativa è data, analogamente, da...

XR = X - 24 * int (X / 24)

Ora, se **Inizio** è l'indirizzo del primo sprite del nostro blocco, riunendo i risultati trovati si trova l'indirizzo del byte contenente il bit che ci interessa:

Loc = Inizio + 64 * Xl * Riga + 64 * Sprite + 3 * Yr + Int (xr / 8)

YR va moltiplicato per **tre** perchè ogni sprite è costituito da **21** righe di tre bytes ciascuna.

Dato inoltre che...

0 < = XR < = 24

...ne consegue che **XR** va diviso per **otto** per sapere in quale dei tre bytes di ordinata relativa **YR** si trova il punto.

Infine...

MASK = 2 ^ (7 - (X and 7))

...fornisce la maschera necessaria per accendere o spegnere il bit.

La tecnica adoperata

Le divisioni sono state realizzate in linguaggio macchina semplicemente tramite sottrazioni successive. Il divisore, cioè, viene sottratto al dividendo contenuto nell'accumulatore, fino a quando quest'ultimo non contiene una quantità minore del divisore stesso: tale quantità è il resto della divisione.

Un indice conta le sottrazioni effettuate fornendo il quoziente.

Il settaggio dei pixels relativi al singolo carattere avviene per colonne, cioè con un movimento dall'alto verso il basso e da sinistra verso destra. Ovviamente la stringa, nel suo complesso, viene riportata per righe, da sinistra verso destra e dall'alto verso il basso.

In pratica

Dopo le spiegazioni teoriche veniamo all'uso pratico delle routines, a cominciare da quella che denomineremo di scrittura.

Il formato è il seguente:

Sys xxxx, Bank, Fspr, Num, Xl, X0, Y0, W, H, Fx, Fy, A$ [,carset]

...in cui:

VIETATO AI MINORI

E' probabile che i neo-utenti di computer non riescano a ben comprendere l'utilità della procedura software descritta nelle presenti pagine. Questa è infatti destinata a coloro che sono già padroni delle principali tecniche di programmazione e desiderano approfondire il modo in cui un elaboratore organizza i vari dati all'interno della memoria. Il programma, pertanto, rappresenta un invito ai lettori più esperti, soprattutto a coloro che, considerando il listato come una base di partenza, riescano a pervenire a procedure più interessanti e capaci, magari, di offrire applicazioni di più ampio respiro. Se hai incominciato da poco, comunque, non scoraggiarti! Tutti coloro che, oggi, vantano una particolare competenza nel campo dell'informatica hanno iniziato con un banale Print "Pippo". Perchè non dovresti riuscire anche tu?...

BANK: numero del banco di memoria all'interno del quale si intende operare.

BANK = **0**: 0 -16383
BANK = **1**: 16384 - 32767
BANK = **2**: 32768 - 49151
BANK = **3**: 49152 - 65535

Valori superiori a questi generano errore.

FSPR: numero (fra zero e 255) del primo sprite interessato dalla nostra routine.

NUM: numero di sprites (da zero a 255) che, a partire da FSPR, vengono interessati dalla nostra routine.

In condizioni normali tale valore non dovrebbe essere superiore a otto.

Ovviamente la routine richiede l'introduzione di questi primi parametri per evitare di influenzare aree di memoria dedicate ad altri scopi.

XL: numero di sprites per riga (da 1 a 8).

X0, Y0: coordinate iniziali (fra zero e 255) della scritta da riportare negli sprites.

W, H: dimensioni in pixels (fra 0 e 255, con lo zero che ha l'effetto di un 256) della finestra all'interno della quale viene riportata la stringa. Tali parametri sono stati introdotti per facilitare la tabulazione della scritta.

Inoltre, se la finestra è piccola è possibile disporre di pennelli di varie dimensioni con cui disegnare o cancellare, usando lo spazio negativo o positivo come unico carattere.

FX, FY: fattori di ingrandimento orizzontale e verticale per la scritta (fra zero e 255, con lo zero che ha l'effetto di un 256).

A$: stringa da riportare nel blocco di sprites.

CARSET: facoltativo, byte alto dell'indirizzo iniziale del set di caratteri.

Può essere usato in riferimento a set ridefiniti o a quelli già disponibili:

Le due routines proposte sono interamente rilocabili e non rischiano, quindi, di entrare in "conflitto" con altri programmi in l.m.

Un piccolo dimostrativo è inserito nelle stesse routine che devono esser digitate, inutile dirlo, con la massima attenzione

Carset = **208** : set maiuscolo-grafici/positivo (default)

Carset = **212** : set maiuscolo-grafici/negativo

Carset = **216** : set minuscolo-maiuscolo/positivo

Carset = **220** : set minuscolo-maiuscolo/negativo

La seconda routine

La seconda routine è stata realizzata soprattutto a sostegno della prima, dato che si limita a posizionare un blocco di sprites a partire da determinate coordinate.

Chiamiamola dunque routine di posizionamento.

Il formato è:

Sys Xxxx, X, Y, Num, Xl, Stepx, Stepy [,fspr]

X, Y: coordinate (ascissa e ordinata) del primo sprite in alto a sinistra del blocco. 0<=Y<=255; 0<=X<=65535 (ma valori di X maggiori di 511 rappresentano una ripetizione)

NUM: numero di sprites costituenti il blocco. 1<=NUM<=8.

XL: numero di sprites per riga. 1<=XL<=8

Stepx: differenza fra l'ascissa di uno sprite e quella del precedente sulla stessa riga. 0<=Stepx<=255 , normalmente dovrebbe valere 24 per sprites non espansi, 48 per sprites espansi.

Stepy: differenza fra l'ordinata di uno sprite e quella del precedente sulla stessa colonna. 0<=Stepy<=255, normalmente dovrebbe valere 21 per sprites non espansi, 42 per sprites espansi.

Fspr: facoltativo, numero del primo sprite, visualizzato in alto a sinistra. 0<=Fspr<=255, i valori da Fspr a Fspr+num -1 vengono posti nelle locazioni da 2040 a 2040+Num-1. In assenza di tale parametro il contenuto delle suddette locazioni resta invariato.

Il caricatore Basic delle due routines è corredato di un breve dimostrativo, che riporta stringhe e disegni in otto sprites variamente disposti.

COPIALO PER TELEFONO

Anche i listati presenti in queste pagine possono esser tirati giù per mezzo del modem; se, ovviamente, ne possedete uno.
La procedura per collegarsi con la nostra banca dati (attiva 24 ore su 24) è riportata su altra parte della rivista.
Chi non possiede il modem (che aspettate a procurarvelo?) può tuttavia richiedere il dischetto, contenente il software, presso il nostro servizio arretrati.

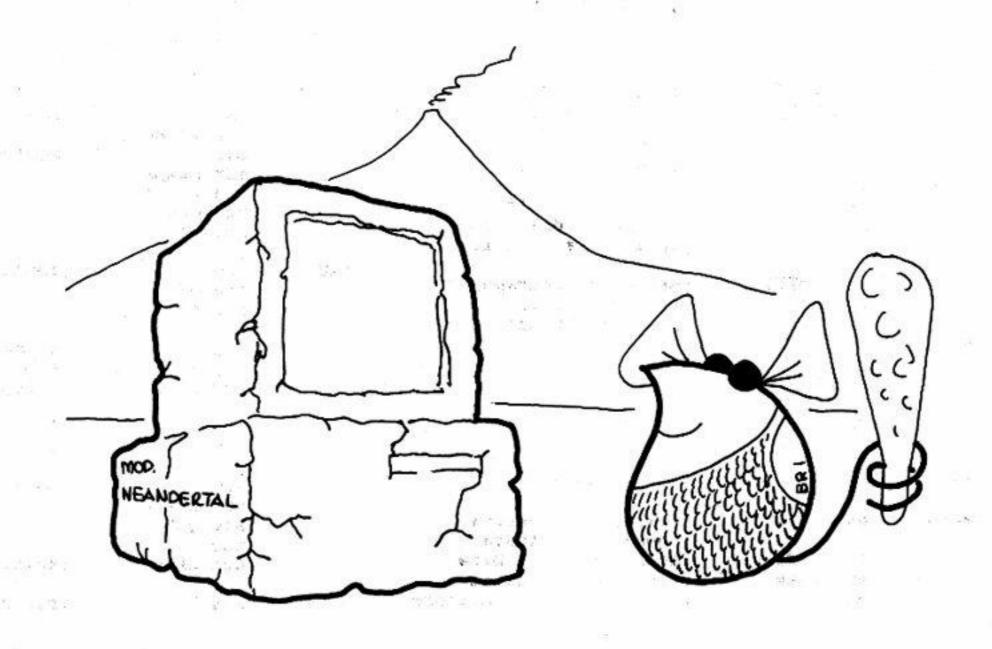
```
100 REM ********************************
110 REM *    COME GESTIRE IN BLOCCO    *
120 REM *     OTTO O PIU' SPRITES      *
130 REM *                              *
140 REM *       DI TITO BRASOLIN       *
150 REM *             1990             *
160 REM ********************************
170 :
180 POKE 55,0: POKE 56,48
190 POKE 53281,6: POKE 53280,6: PRINT CHR$(159)
200 GOSUB 560
210 GOSUB 730
220 GOSUB 860
230 PRINT CHR$(147): PRINT " VUOI UNA DIMOSTRAZIONE (S/N) ? "
240 GET A$: IF A$="S" THEN 290
250 IF A$<>"N" THEN 240
260 END
270 :
280 REM * DEMO *
290 VI=53248: PRINT CHR$(147)
300 POKE VI+21,255: POKE VI+23,255: POKE VI+29,255
310 FOR R=0 TO 7:POKE VI+39+R,15: NEXT R
320 SYS S1,0,192,8,4,0,0,96,42,96,42," "
330 SYS S1,0,192,8,4,0,0,96,42,1,1," COMMODORE    COMPUTER       CLUB    ",212
340 SYS S1,0,192,8,4,0,24,96,18,1,2,"PRESENTA....",216
350 FOR R=255 TO 48 STEP -1
360 SYS S2,152,166,8,4,R,R-6,192
370 NEXT R
380 FOR A=0 TO 2000: NEXTA
390 SYS S2,24,50,8,3,48,42
400 SYS S1,0,192,8,3,0,0,72,63,72,63," ",212
410 SYS S1,0,192,8,3,0,39,48,24,1,1,"SENO  E     COSENO"
420 FOR W=1 TO 8
```

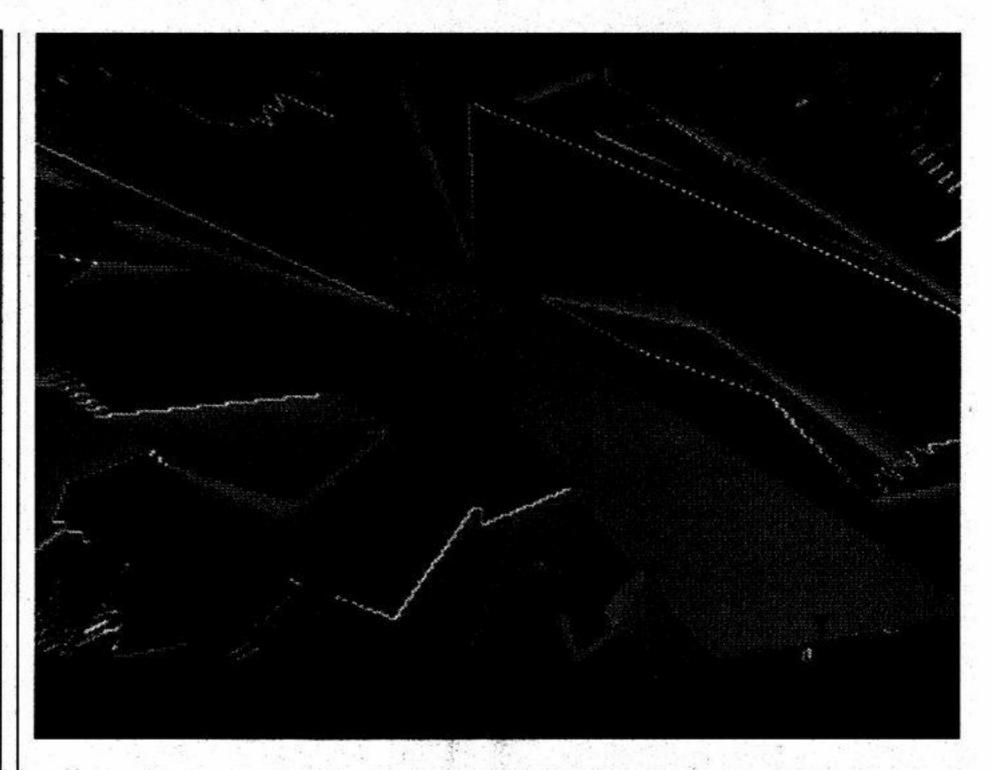
```
430 FOR X=-π TO +π STEP π/24
440 X1=36+X*(36/π)
450 Y1=17+17*SIN(X): Y2=17+17*COS(X)
460 SYS S1,0,192,8,3,X1,Y1,W,W,1,1," "
470 SYS S1,0,192,8,3,X1,Y2,W,W,1,1,"■",212
480 NEXT X
490 NEXT W
500 FOR A=0 TO 2000: NEXT A
510 SYS S2,136,66,8,2,48,42
520 FORR=1TO48:SYS S1,0,192,8,2,0,0,R,84,6,2,"CIAO● ":NEXTR
530 END
540 :
550 REM * TITOLI *
560 PRINT CHR$(147): PRINT " PROGRAMMA PER LA GESTIONE DI SPRITES   ";
570 PRINT " AFFIANCATI, COME SE FOSSERO UNA PAGINA ";
580 PRINT " GRAFICA . "
590 PRINT : PRINT " VI SONO DUE ROUTINES INDIPENDENTI       ";
600 PRINT " E RILOCABILI: "
610 PRINT: PRINT
620 PRINT "ROUTINE 'SCRITTURA'  (389 BYTES) "
630 PRINT: PRINT "SYS XXXX,BANK,FSPR,NUM,XL,X0,Y0,W,H,FX, ";
640 PRINT "            FY,A$ [,CARSET]"
650 PRINT: PRINT
660 PRINT "ROUTINE 'POSIZIONAMENTO'  (153 BYTES) "
670 PRINT: PRINT "SYS XXXX,X,Y,NUM,XL,STEPX,STEPY [,FSPR] "
680 PRINT: PRINT " PER I PARTICOLARI, LEGGI SU C.C.C.     "
690 PRINT CHR$(18)" PREMI UN TASTO ... "
700 POKE 198,0: WAIT 198,1: GET A$: RETURN
710 :
720 REM * INPUT INDIRIZZI INIZIALI *
730 PRINT CHR$(147): PRINT " DIMMI L'INDIRIZZO INIZIALE DI CIASCUNA  ROUTINE "
740 S1=32768
750 PRINT: PRINT " ROUTINE 'SCRITTURA'       ";S1
760 PRINT CHR$(145);TAB(25);: INPUT S1
770 S2=S1+389
780 PRINT: PRINT " ROUTINE 'POSIZIONAMENTO' ";S2
790 PRINT CHR$(145);TAB(25);: INPUT S2
800 IF (S2+152<S1) OR (S2>S1+388) THEN 830
810 PRINT CHR$(18)" LE ROUTINES NON SI POSSONO ACCAVALLARE "
820 FOR A=0 TO 2000: NEXT A: GOTO 730
830 RETURN
840 :
850 REM * LETTURA DATI PER L.M. *
860 PRINT CHR$(147)"ATTENDI L'ALLOCAZIONE DELLE DUE ROUTINES"
870 C=0
880 FOR A=S1 TO S1+388
890 READ B: POKE A,B: C=C+B: NEXT A
900 IF C<>44701 THEN PRINT " ERRORE NEI DATI DELLA PRIMA ROUTINE ": STOP
910 C=0
920 FOR A=S2 TO S2+152
930 READ B: POKE A,B: C=C+B: NEXT A
940 IF C<>19290 THEN PRINT " ERRORE NEI DATI DELLA SECONDA ROUTINE ": STOP
950 RETURN
960 :
970 :
980 :
990 REM * DATI ROUTINE SCRITTURA
1000 DATA 032,241,183,224,004,144,003,076,072,170,142,019,003,032,241,183
1010 DATA 142,052,003,032,241,183,142,053,003,032,241,183,142,054,003,202
1020 DATA 224,008,176,227,032,241,183,142,055,003,032,241,183,142,056,003
1030 DATA 032,241,183,138,024,109,055,003,141,057,003,032,241,183,138,024
1040 DATA 109,056,003,141,058,003,032,241,183,142,252,003,134,251,032,241
1050 DATA 183,142,253,003,032,253,174,032,158,173,032,130,183,140,059,003
1060 DATA 164,034,132,247,164,035,132,248,162,208,032,121,000,201,044,208
1070 DATA 003,032,241,183,142,254,003,120,165,001,041,248,133,001,160,000
1080 DATA 169,128,133,253,173,055,003,133,249,041,007,170,169,128,202,048
1090 DATA 003,074,144,250,133,254,204,059,003,240,094,132,002,177,247,016
1100 DATA 014,201,255,208,004,169,094,208,016,041,127,009,064,208,010,201
1110 DATA 096,176,004,041,191,144,002,041,223,133,034,169,000,240,066,166
1120 DATA 250,232,236,058,003,240,009,198,252,208,078,200,192,008,208,068
1130 DATA 070,254,144,002,102,254,164,002,198,251,208,012,173,252,003,133
1140 DATA 251,070,253,144,003,102,253,200,230,249,165,248,205,057,003,208
1150 DATA 165,142,056,003,236,058,003,208,139,165,001,009,008,133,001,088
1160 DATA 096,006,034,042,006,034,042,006,034,042,109,254,003,133,035,160
1170 DATA 000,174,056,003,173,253,003,133,252,134,250,138,162,000,201,021
1180 DATA 144,005,233,021,232,208,247,133,020,134,021,165,249,162,000,201
```

```
1190 DATA 024,144,005,233,024,232,208,247,236,054,003,176,130,074,074,074
1200 DATA 024,101,020,006,020,101,020,133,020,138,174,054,003,101,021,202
1210 DATA 208,251,205,053,003,176,228,109,052,003,133,021,173,019,003,105
1220 DATA 000,074,102,021,106,102,021,106,005,020,133,020,177,034,036,259
1230 DATA 240,008,165,254,001,020,129,020,208,008,165,254,073,255,033,020
1240 DATA 129,020,056,176,182
1250 :
1260 REM * DATI ROUTINE POSIZIONAMENTO
1270 DATA 032,253,174,032,235,183,134,253,032,241,183,134,254,202,224,008
1280 DATA 144,003,076,072,178,032,241,183,142,019,003,202,224,008,176,242
1290 DATA 032,241,183,142,052,003,032,241,183,142,053,003,032,121,000,201
1300 DATA 044,208,017,032,241,183,138,162,000,157,248,007,056,105,000,232
1310 DATA 228,254,208,245,006,254,160,001,132,002,136,165,021,133,252,165
1320 DATA 020,133,251,174,019,003,165,252,041,001,240,010,165,002,013,016
1330 DATA 208,141,016,208,208,010,165,002,073,255,045,016,208,141,016,208
1340 DATA 006,002,165,251,153,000,208,024,109,052,003,133,251,144,002,230
1350 DATA 252,200,165,253,153,000,208,200,196,254,240,012,202,208,199,024
1360 DATA 109,053,003,133,253,024,144,179,096
1370 END
```

```
;****************************************
;
;    stampa una stringa in un gruppo
;            di sprites
;
;        routine rilocabile
;         di tito brasolin
;
;              1990
;
;****************************************

            ;
chkcom      = $aefd
frmevl      = $ad9e
len1        = $b782
chrgot      = $79
            ;
carnum      = $02
loc1        = $14
loc2        = $22
stradr      = $f7
x           = $f9
y           = $fa
cntfx       = $fb
cntfy       = $fc
test        = $fd
mask        = $fe
            ;
bank        = $0313
fsprite     = $0334
num         = $0335
x1          = $0336
x0          = $0337
y0          = $0338
width       = $0339
height      = $033a
len         = $033b
fx          = $03fc
fy          = $03fd
carset      = $03fe
mode        = $03ff
            ;
;---------------------------------------
            ;
            jsr $b7f1       ;leggi banco
            cpx #$04        ;errore se >3
            bcc noerr
            ;
errore      jmp $b248       ;illegal quantity
            ;
noerr       stx bank
            ;
            jsr $b7f1       ;leggi il numero
            stx fsprite     ;del primo sprite
            ;
            jsr $b7f1       ;leggi il numero di
            stx num         ;sprites interessati
            ;
            jsr $b7f1       ;leggi numero di sprites
            stx x1          ;per riga
            dex             ;errore se >8
            cpx #$08
            bcs errore
            ;
            jsr $b7f1       ;leggi ascissa
            stx x0
            ;
            jsr $b7f1       ;leggi ordinata
            stx y0
            ;
            jsr $b7f1       ;leggi larghezza
            txa             ;della finestra
            clc
            adc x0
            sta width
            ;
            jsr $b7f1       ;leggi altezza
            txa             ;della finestra
            clc
            adc y0
            sta height
            ;
            jsr $b7f1       ;leggi i fattori
            stx fx          ;di ingrand.
            stx cntfx
            ;               ;orizzontale
            jsr $b7f1       ;e verticale
            stx fy
            ;
            jsr chkcom      ;leggi i parametri
            jsr frmevl      ;relativi alla
            jsr len1        ;stringa da riportare
            sty len
            ldy loc2
            sty stradr
            ldy loc2+1
            sty stradr+1
            ;
            ldx #$d0        ;se presente, leggi
            jsr chrgot      ;il byte alto
            cmp #$2c        ;dell'indirizzo
            bne ok1         ;iniziale del set di
                            ;caratteri
            jsr $b7f1       ;default = #$d0
            ;
ok1         stx carset
            ;
```

```
;-----------------------------------------
        ;
        sei             ;blocca irq
        lda $01         ;scopri generatore
        and #%11111001  ;caratteri e ram sotto
        sta $01         ;il s.o.
        ;
;-----------------------------------------
        ;
        ldy #$00        ;primo carattere
        ;
        lda #%10000000  ;inizializza maschera
        sta test        ;per 'testare' i bits
        ;
nextb   lda x0          ;inizializza ascissa
        sta x
        and #%00000111  ;crea maschera per
        tax             ;accendere o spegnere
        lda #%10000000  ;i pixels nello sprite
        ;
next1   dex
        bmi out2
        lsr a
        bcc next1
        ;
out2    sta mask
        ;
nexta   cpy len         ;finita stringa?
        beq fine        ;se si', fine programma
        sty carnum
        ;
        lda (stradr),y  ;leggi codice ascii
        bpl ok3         ;e traducilo
        cmp #255        ;in codice schermo
        bne ok4
        lda #94
        bne out1
        ;
ok4     and #%01111111
        ora #%01000000
        bne out1
        ;
ok3     cmp #96
        bcs ok5
        and #%10111111
        bcc out1
        ;
ok5     and #%11011111
        ;
out1    sta loc2
        lda #$00
        beq cont        ;salto incondizionato
        ;
;-----------------------------------------
        ;
out     ldx y           ;incrementa ordinata
        inx             ;raggiunto il limite inf.
        cpx height      ;della finestra?
        beq skip1
        ;
        dec cntfy       ;finito l'ingr.vert.
        bne nexty       ;di un pixel?
        ;
        iny             ;se si',passa al
        cpy #$08        ;prossimo byte del
        bne nxtbyt      ;carattere
        ;
skip1   lsr mask        ;aggiorna maschera
        bcc skip3
        ror mask
        ;
skip3   ldy carnum
        dec cntfx       ;finito ingr.orizz.
        bne skip5       ;di una 'colonna'
                        ;di 8 pixels?
        lda fx          ;se si',inizializza
        sta cntfx       ;contatore e aggiorna
        lsr test        ;la maschera per testare
        bcc skip5
        ;
        ror test        ;se finito carattere,
        iny             ;aggiorna puntatore al
                        ;carattere in corso
skip5   inc x           ;incrementa ascissa
        lda x           ;raggiunto limite destro
        cmp width       ;della finestra?
        bne nexta       ;se no, prosegui
                        ;se si',reinizializza
        stx y0          ;ordinata
        cpx height      ;raggiunto limite inf.?
        bne nextb       ;se no, prosegui
                        ;se si', fine programma
;-----------------------------------------
        ;
fine    lda $01         ;copri generatore
        ora #%00000110  ;caratteri
        sta $01
        cli             ;abilita irq
        rts             ;ritorna
        ;
;-----------------------------------------
        ;
cont    asl loc2        ;calcola l'indirizzo
        rol a           ;base della matrice
        asl loc2        ;del carattere
        rol a
        asl loc2
        rol a
        adc carset
        sta loc2+1
        ;
;-----------------------------------------
        ;
nextx   ldy #$00        ;primo byte del carattere
        ;
        ldx y0          ;inizializza ordinata
        ;
nxtbyt  lda fy          ;inizializza counter per
        sta cntfy       ;ingr. verticale
        ;
nexty   stx y           ;aggiorna ordinata
        txa
        ldx #$00
        ;
next2   cmp #21         ;calcola int(y/21)
        bcc out3
        sbc #21
        inx
        bne next2
        ;
out3    sta loc1
        stx loc1+1
        ;
        lda x
        ldx #$00
        ;
next3   cmp #24         ;calcola int(x/24)
        bcc out4
        sbc #24
        inx
        bne next3
        ;
out4    cpx x1          ;verifica che l'ascissa
jump    bcs out         ;non sia troppo alta
        lsr a           ;calcola :
        lsr a           ;(x-24*int(x/24))/8...
        lsr a
        clc
        adc loc1        ;+3*(y-21*int(y/21))
```

```
        asl loc1
        adc loc1
        sta loc1
        txa
        ldx x1
        ;
next4   adc loc1+1      ;int(x/24)+x1*int(y/21)
        dex
        bne next4
        cmp num         ;controlla se sei uscito
        bcs jump        ;dal range di sprites
        adc fsprite     ;somma primo sprite
        sta loc1+1
        lda bank        ;somma banco
        adc #$00        ;e completa i calcoli
        lsr a
        ror loc1+1
        ror a
        ror loc1+1
        ror a
        ora loc1
        sta loc1
        lda (loc2),y    ;controlla il bit
        bit test        ;del carattere in corso
        beq spegni      ;e se e' acceso...
        ;
        lda mask        ;...accendine uno
        ora (loc1,x)    ;nello sprite
        sta (loc1,x)
        bne jump2
        ;
spegni  lda mask        ;altrimenti spegnilo
        eor #$ff
        and (loc1,x)
        sta (loc1,x)
jump2   sec
        bcs jump        ;vai ai controlli finali
        ;
;----------------------------------------
```

```
;***************************************
;
;   posiziona un blocco di sprites
;
;          di tito bresolin
;
;                1989
;
;       disassemblato commentato
;
;formato: sys xxxx,x,y,num,xl,stepx,
;                  stepy [,sprnum]
;
;***************************************
         ;
chrgot   = $79
chkcom   = $aefd
getnum   = $b7eb
         ;
mask     = $02
x0       = $14
x        = $fb
y0       = $fd
num      = $fe
xl       = $0313
stepx    = $0334
stepy    = $0335
         ;
;---------------------------------------
         ;
         jsr chkcom      ;leggi coordinate
         jsr getnum
         stx y0
         ;
         jsr $b7f1       ;leggi numero di
         stx num         ;sprites interessati
         dex             ;max=8
         cpx #$08
         bcc ok1
         ;
errore   jmp $b248       ;illegal quantity
         ;
ok1      jsr $b7f1       ;leggi numero di
         stx xl          ;sprites per riga
         dex             ;max=8
         cpx #$08
         bcs errore
         ;
         jsr $b7f1       ;leggi il 'passo'
         stx stepx       ;orizzontale...
         ;
         jsr $b7f1
         stx stepy       ;...e verticale
         ;
         jsr chrgot      ;se presente, leggi
         cmp #$2c        ;il numero del primo
         bne main        ;sprite visualizzato
         ;
         jsr $b7f1       ;setta opportunamente
         txa             ;i puntatori alla
         ldx #$00        ;definizione sprites
next1    sta $07f8,x
         sec
         adc #$00
         inx
         cpx num
         bne next1
         ;
;---------------------------------------
         ;
main     asl num         ;(bisogna impostare
                         ; num*2 coordinate)
         ldy #$01        ;maschera per i bits
         sty mask        ;alti delle ascisse
         dey             ;l'indice y e' puntatore
                         ;alla coordinate
         ;
next3    lda x0+1        ;inizializza ascisse
         sta x+1
         lda x0
         sta x
         ;
         ldx xl
         ;
next2    lda x+1         ;controlla se
         and #%00000001  ;ascissa>255
         beq spegni      ;e imposta di
         ;               ;conseguenza il bit
         lda mask        ;alto dell'ascissa
         ora $d010       ;dello sprite
         sta $d010       ;in questione
         bne go1
         ;
spegni   lda mask
         eor #%11111111
         and $d010
         sta $d010
         ;
go1      asl mask
         ;
         lda x           ;imposta ascissa
         sta $d000,y
         clc             ;aggiorna ascissa
         adc stepx
         sta x
         bcc ok2
         inc x+1
         ;
ok2      iny             ;imposta ordinata
         lda y0
         sta $d000,y
         ;
         iny             ;prossimo sprite
         cpy num         ;abbiamo finito?
         beq fine        ;salta alla fine
         ;
         dex             ;finita riga
         bne next2       ;di sprites?
         ;
         clc             ;aggiorna ordinata
         adc stepy
         sta y0
         clc
         bcc next3       ;salto incondizionato
         ;
fine     rts             ;ritorna
```

Il breve programma presentato è comprensibile dai lettori più esperti, ma anche i principianti possono digitarlo e osservarne i simpatici effetti

UNA... ZEBRA A COLORI PER IL C/64

Spesso servono presentazioni ricche di effetti coloratissimi; provate a visualizzare un. "passaggio pedonale" in technicolor

di Andrea Fasce

Come molti sapranno, lo schermo viene disegnato, linea per linea, da un cannone elettronico (il **Raster**) che "spara" i punti luminosi sullo schermo da sinistra verso destra e dall'alto verso il basso.

Ogni linea orizzontale dello schermo ha però una durata ben precisa a seconda dei casi, perciò bisogna registrare in memoria la durata di ogni linea di schermo in modo che il programma sappia quanto dura una determinata linea di schermo.

(Per approfondire l'argomenrtio si consiglia di rileggere i numeri di C.C.C. n. 58 e n. 68)

COPIALO PER TELEFONO

Anche i listati presenti in queste pagine possono esser tirati giù per mezzo del modem; se, ovviamente, ne possedete uno.
La procedura per collegarsi con la nostra banca dati (attiva 24 ore su 24) è riportata su altra parte della rivista.
Chi non possiede il modem (che aspettate a procurarvelo?) può tuttavia richiedere il dischetto, contenente il software, presso il nostro servizio arretrati.

Il Programma

Il programma proposto potrà risultare molto utile in più di una situazione e, in particolare, nel caso di presentazioni di propri programmi.

Il suo scopo è, infatti, di creare tante linee colorate che "scrollano" verso l'alto di un pixel per volta (in modo **ACE**, cioè senza scatti e in modo fluido) in un ciclo infinito.

I valori di ritardo sono allocati da **$C200** (#49664)e i colori in un buffer allocato da **$C100** (#49408); la routine parte invece da **$C000** (#49152).

E' bene tener presente che l'alterazione della locazione C200 è riservata ai lettori più esperti dal momento che la modifica del suo contenuto non provvede a modificare la velocità di scorrimento, ma presenta effetti collaterali.

E' invece possibile modificare la tabella per avere strisce di spessore più o meno diverso.

E' stata anche inserita una routine per disabilitare gli interrupt (presa da n. 68 di C.C.C.), molto utile nel caso si usi il programma *Megaraster* in programmi in basic perchè rallenta di molto l'esecuzione della macchina.

Difetti e pregi di *Megaraster*

Il programma provvede, ad ogni Interrupt, a disegnare le linee una per una, e a far scrollare di un posto i colori nel buffer.

Per questo si possono riscontrare *enormi* rallentamenti nel caso di programmi in basic.

Un altro difetto, presente solo in caso di presenza di sprite, consiste nel vedere sfarfallare la zona video ad essi vicina (è un bug del VIC).

Non è stata implementata la gestione anche del bordo per semplicità, in quanto sarebbe servito manipolare anche il byte alto del raster register, con conseguente ulteriore perdita di tempo e sfarfallamenti vari.

Sono però stati ridotti al minimo gli sfarfallamenti durante gli scroll dello schermo verso l'alto (quando si preme il tasto Crs / Down).

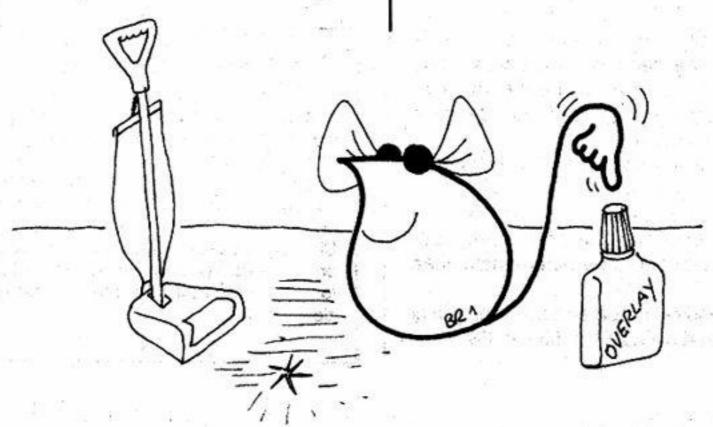
VIETATO AI MINORI

E' probabile che i neo-utenti di computer non riescano a ben comprendere l'utilità della procedura software descritta nelle presenti pagine. Questa è infatti destinata a coloro che sono già padroni delle principali tecniche di programmazione e desiderano approfondire il modo in cui un elaboratore organizza i vari dati all'interno della memoria. Il programma, pertanto, rappresenta un invito ai lettori più esperti, soprattutto a coloro che, considerando il listato come una base di partenza, riescano a pervenire a procedure più interessanti e capaci, magari, di offrire applicazioni di più ampio respiro. Se hai incominciato da poco, comunque, non scoraggiarti! Tutti coloro che, oggi, vantano una particolare competenza nel campo dell'informatica hanno iniziato con un banale Print "Pippo". Perchè non dovresti riuscire anche tu?...

Programma caricatore in Basic

Il programma in basic carica in memoria la routine in LM, i tempi e (a scelta) i colori (tatnte barre orizzontali colorate). Molte volte si hanno però effetti migliori evitando di scrivere i colori in memoria. Il programma in basic e la routine in LM sono stati scritti volutamente facili in modo da farli comprendere anche ai meno esperti.

Il disassemblato Im

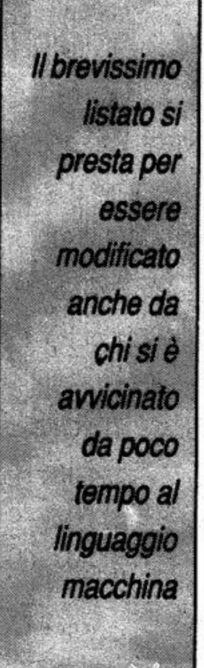
Il programma in LM è molto facile da capire ed è simile a molti altri programmi che gestiscono il Raster. La parte più interessante è sicuramente quella riferita alla gestione del buffer dei colori. Questo contiene duecento colori (equivalenti alle righe visibili dello schermo); il programma legge dapprima il primo (scusate il gioco di parole) colore del buffer e lo trasferisce nello Stack. Poi provvede a spostare tutti i colori contenuti nel buffer di un posto verso sinistra; a questo punto rimane libero l'ultimo byte del buffer, che sarà riempito con il colore precedentemente salvato nello Stack.

```
10 rem ********************************
20 rem *    m e g a r a s t e r       *
30 rem ********************************
40 rem * (c) 1990 by:                 *
50 rem * andrea fasce - genova        *
60 rem ********************************
70 :
100 te=49664:ck=0
110 printchr$(147)
120 printtab(13);"***************"
130 printtab(13);"* megaraster *"
140 printtab(13);"* by: andrea *"
150 printtab(13);"*     fasce  *"
160 printtab(13);"***************"
170 print:print"attendi...."
180 :
190 print:print"scrivo routine lm"
200 fora=49152to49277
210 readb:ck=ck+b:pokea,b:next
220 if ck<>15657thenprint"errore":end
230 :
240 print:print"scrivo ritardi"
250 fora=tetote+255:pokea,8:next
260 pokete,14:fora=te+1tote+255step8
265 pokea,1:next
270 :
280 print:print"scrivo colori (s/n)"
285 geta$:ifa$=""then285
286 ifa$<>"s"then450
290 z=49408:y=0:gosub900
300 z=49421:y=1:gosub900
310 z=49437:y=2:gosub900
320 z=49453:y=3:gosub900
330 z=49469:y=4:gosub900
340 z=49485:y=5:gosub900
350 z=49501:y=6:gosub900
360 z=49517:y=7:gosub900
370 z=49533:y=8:gosub900
380 z=49549:y=9:gosub900
390 z=49565:y=10:gosub900
400 z=49581:y=11:gosub900
410 z=49597:y=12:gosub900
420 z=49613:y=13:gosub900
430 z=49629:y=14:gosub900
440 z=49645:y=15:gosub900
450 printchr$(147)
460 print"attivare con sys 49152"
470 print"disattivare con sys49249"
480 print:print"premi un tasto"
490 poke198,0:wait198,1
500 sys49152:end
510 end :end :end :end :end
520 rem (c) 1990 by: andrea fasce
900 fora=ztoz+16
910 pokea,y:next
920 return
1000 data 120,169,036,141,020,003,169
1001 data 192,141,021,003,169,049,141
1002 data 018,208,173,017,208,041,127
1003 data 141,017,208,169,000,141,014
1004 data 220,169,001,141,026,208,088
1005 data 096,169,001,141,025,208,162
1006 data 000,189,000,193,188,000,194
1007 data 136,208,253,141,033,208,141
1008 data 033,208,232,224,200,208,236
1009 data 173,000,193,072,162,001,189
1010 data 000,193,202,157,000,193,232
1011 data 232,224,200,208,243,104,141
1012 data 199,193,169,000,141,032,208
1013 data 141,033,208,076,049,234,120
1014 data 169,000,141,018,208,169,049
1015 data 141,020,003,169,234,141,021
1016 data 003,169,001,141,014,220,169
1017 data 240,141,026,208,088,096,000
1018 end

ready.
```

```
; ***********************
; *                     *
; *     megaraster      *
; *         by          *
; *    andrea fasce     *
; *                     *
; ***********************
;
raster=$d012
irq=$ea31
hirast=$d011
timer=$dc0e
vicirq=$d01a
vicflg=$d019
bordo=$d020
video=$d021
buffer=$c100
tempi=$c200
;
*=49152
       sei              ; setta interrupt
       lda #<main       ; punta
       sta $0314        ; l' interrupt
       lda #>main       ; a main
       sta $0315        ;
       lda #$31         ; setta linea
       sta raster       ; raster a $31
       lda hirast       ; resetta
       and #$7f         ; hi-byte
       sta hirast       ; del raster
       lda #0           ; spegne il
       sta timer        ; timer
       lda #1           ; setta interrupt
       sta vicirq       ; video (vic)
       cli              ; resetta interrupt
       rts              ; esce
;
main   lda #$01         ; conferma
       sta vicflg       ; il raster-irq
       ldx #0           ; contatore #0
loop   lda buffer,x     ; legge buffer
       ldy tempi,x      ; legge ritardo
lp1    dey              ; esegue il
       bne lp1          ; ritardo
       sta video        ; scrive colore
       sta video        ; in schermo
       inx              ; incrementa contatore
       cpx #$c8         ; contatore=$c8
       bne loop         ; se no vai a lp1
       lda buffer       ; se si carica buffer
       pha              ; scrivilo nello stack
       ldx #1           ; contatore=#1
lp2    lda buffer,x     ; carica buffer
       dex              ; decrementa contatore
       sta buffer,x     ; riscrivi buffer
       inx              ; incrementa contatore
       inx              ; incrementa contatore
       cpx #$c8         ; contatore=$c8
       bne lp2          ; se no vai a lp2
       pla              ; se si recupera buffer
       sta buffer+199   ; scrivi buffer in buffer+199
       lda #0           ; accumulatore=#0
       sta bordo        ; scrive in bordo e video
       sta video        ; per evitare sfarfallamenti
       jmp irq          ; esci
;
; routine reset raster-interrupt
;
;
reset  sei              ; setta interrupt
       lda #0           ; rimette a posto
       sta raster       ; il raster-register
       lda #$31         ; i registri
       sta $0314        ; di interrupt
       lda #$ea         ;
       sta $0315        ;
       lda #$01         ; setta il
       sta timer        ; timer
       lda #$f0         ; setta
       sta $d01a        ; vic-interrupt
       cli              ; resetta interrupt
       rts              ; esce
       .end
```

CIOMP!

FR FERROVIE RATTESCHE

LINEA RAT CITY RATTONIA KM. 124

Il brevissimo listato si presta per essere modificato anche da chi si è avvicinato da poco tempo al linguaggio macchina

UN MESSAGGIO LAMPEGGIANTE

Un brevissimo programma che, nonostante sia scritto in linguaggio macchina, può essere usato anche da un principiante

di Andrea Fasce

Supercolor è uno di quei programmi utili per ottenere effetti insoliti e, comunque, difficilmente ottenibili in Basic a causa della lentezza intrinseca del linguaggio interprete.

Il suo scopo è quello di far cambire rapidamente colore ad alcune locazioni di schermo (nel nostro caso le **prime quattro righe** video) in modo di ottenere un effetto interessante sopratutto durante le presentazioni di programmi, in quanto il titolo comparirà per dissolversi, poi, lentamente.

Il programma

Il programma scambia velocemente i colori dei caratteri che compaiono nelle prime quattro righe di schermo.

Si passa dal nero al bianco, attraverso le tre gradazioni di grigio disponibili, e poi nuovamente dal bianco al nero, sempre passando "attraverso" i grigi.

COPIALO PER TELEFONO

Anche i listati presenti in queste pagine possono essere tirati giù per mezzo del modem, se, ovviamente, ne possedete uno.
La procedura per collegarsi con la nostra banca dati (attiva 24 ore su 24) è riportata su altra parte della rivista.
Chi non possiede il modem (che aspettate a procurarvelo?) può tuttavia richiedere il dischetto, contenente il software e, presso il nostro servizio arretrati.

VIETATO AI MINORI

E' probabile che i neo-utenti di computer non riescano a ben comprendere l'utilità della procedurqa software descritta nelle presenti pagine. Questa è infatti destinata a coloro che sono già padroni delle principali tecniche di programmazione e desiderano approfondire il modo in cui un elaboratore organizza i vari dati all'interno della memoria.
Il programma, pertanto, rappresenta un invito ai lettori più esperti, soprattutto a coloro che, considerando il listato come una base di partenza, riescano a prevenire a procedure più interessanti e capaci, magari, di offrire applicazioni di più ampio respiro.
Se hai incominciato da poco, non scoraggiarti. Tutti coloro che oggi, vantano una particolare competenza nel campo dell'informatica hanno iniziato con una banale print "Pippo".
Perché non dovresti riuscire anche tu?...

La routine LM in questione è stata scritta in modo facile per farla comprendere anche da chi si è avvicinato da pochissimo al Linguaggio Macchina.

Una nota di rilievo è rappresentata sicuramente da una routine del Kernal del C/64, largamente usata dal programma *Supercolor.*

La routine in questione è quella allocata da **$EEB3** (in decimale #61107) ed esegue un ritardo preciso di 1 microsecondo.

Essa si rivela molto utile in tutti quei programmi che richiedono temporizzazioni molto precise, come musiche in interrupt, effetti raster, sprite e altro.

Il modo migliore per usarla è il seguente:

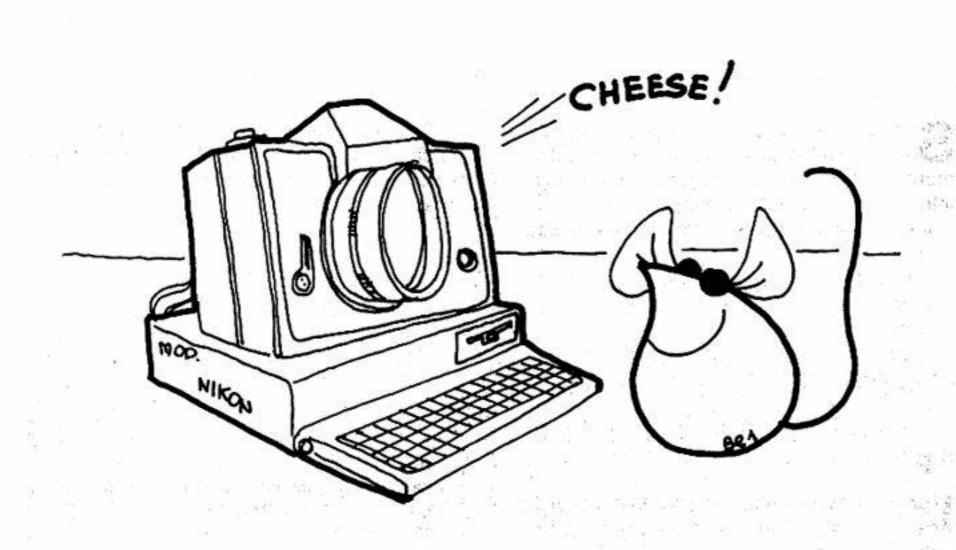
1- caricare il registro X con il numero dei secondi desiderati
2- Saltare alla routine ($EEB3) con JSR
3- Decrementare il registro X
4- Tornare al punto 2 finchè X non è nullo.

Un esempio può essere il seguente:

```
      LDX   #$C8
lp1   JSR   $EEB3
      DEX
      BNE   lp1
```

In questo caso (se si considera trascurabile il tempo per eseguire le istruzioni del programma stesso) il ritardo sarà di $C8 (in decimale #200) microsecondi.

Disassemblato routine da $EEB3:

```
EEB3   TXA
EEB4   LDX   #$B8 ; carica X con $B8
EEB6   DEX
EEB7   BNE   $EEB6 ; se X > 0 vai a
             $EEB6
EEB9   TAX
EEBA   RTS
```

La locazione 49205 contiene il numero di celle video interessate al lampeggio; provate ad alterane il contenuto

```
;supercolor
;programma scritto da andrea fasce
;assemblato con: macroassembler 64
;(c) 1990 by andrea fasce
;
*=49152

main  lda #0          ; carica acc. con nero
      jsr color       ; salta a color
      lda #11         ; carica acc. con grigio 1
      jsr color       ; salta a color
      lda #12         ; carica grigio 2
      jsr color       ; salta a color
      lda #15         ; carica grigio 3
      jsr color       ; salta a color
      lda #1          ; carica bianco
      jsr color       ; salta a color
      lda #15         ; carica grigio 3
      jsr color       ; salta a color
      lda #12         ; carica grigio 2
      jsr color       ; salta a color
      lda #11         ; carica grigio 1
      jsr color       ; salta a color
      lda #0          ; carica nero
      jsr color       ; salta a color
      rts             ; esce
      ;
color ldx #$00        ; contatore=$00
lp1   sta $d800,x     ; scrive colore in memoria colori
      inx             ; incrementa contatore
      cpx #$a0        ; contatore=$a0
      bne lp1         ; se no vai a lp1
      ldx #$90        ; se si contatore=$90
lp2   jsr $eeb3       ; ritardo di 1 microsecondo
      dex             ; decrementa contatore
      bne lp2         ; se non e' zero vai a lp2
      rts             ; ritorna a main
.end
```

```
100 rem ********************
110 rem *    supercolor    *
120 rem *        by        *
130 rem *   andrea fasce   *
140 rem ********************
150 rem
160 poke53280,0:poke53281,0:poke646,1
170 printchr$(147)
180 print:print:print:print
200 print" leggo i data..."
210 gosub400
220 print"premi un tasto"
230 geta$:ifa$=""then230
240 printchr$(147):poke646,0
250 printtab(15)"supercolor"
260 printtab(19)"by"
270 printtab(14)"andrea fasce"
280 printtab(16)"(c) 1990"
290 sys49152:goto290
300 end:end:end:end:end:end:
350 :
400 fora=49152to49216:readb
405 pokea,b:ck=ck+b:next
410 ifck<>7242thenprint"errore":end
420 print" ok":return
1000 data 169,000,032,046,192,169,011
1001 data 032,046,192,169,012,032,046
1002 data 192,169,015,032,046,192,169
1003 data 001,032,046,192,169,015,032
1004 data 046,192,169,012,032,046,192
1005 data 169,011,032,046,192,169,000
1006 data 032,046,192,096,162,000,157
1007 data 000,216,232,224,160,208,248
1008 data 162,144,032,179,238,202,208
1009 data 250,096,-1
1010 end

ready.
```

LA POSTA DEL C/128

(a cura di Domenico Pavone)

STRANI MESSAGGI

Ho notato un comportamento strano del mio C/128 dotato di drive. Se, dopo aver regolarmente acceso il computer e l'unità a disco, spengo quest'ultima mentre fa i suoi soliti "rumorini" iniziali, sul monitor compare questa scritta:

U1 13 0 01 00

Sapete dirmi perchè, e che cosa significa.?

(Luca Cassioli - P.D.Tavole)

Prima ancora che ricercarne un significato tecnico, la comparsa di quei caratteri sullo schermo dovrebbe essere interpretata come un messaggio di "sofferenza" del C/128. Già, perchè non è proprio così ortodosso spegnere il drive nella fase di inizializzazione, a computer acceso: occhio, ci si potrebbe ritrovare in un centro di assistenza in men che non si dica.

Comunque, il messaggio è certamente una sequenza che il computer stava per inviare al drive, e che è stata dirottata sullo schermo dal "trauma" subito. In particolare, la sigla **U1** corrisponde al comando di accesso diretto **Block Read**, che legge il contenuto di un blocco del disco e lo trasferisce in buffer interno del drive.

I valori che seguono svelano di quale incombenza si stava occupando il computer prima del forzato blackout.

La sintassi del comando, normalmente inviato al drive dopo l'apertura del canale comandi (**Open 15, 8, 15**) e di un canale di accesso diretto (per esempio il numero 5 con **Open 5, 8, 5, "#"**) è infatti:

Print #5, "U1; canale; drive; traccia; settore La scritta apparsa sul monitor fa quindi parte della sequenza di avviamento del computer, che prevede uncontrollo della prima traccia di un dischetto (settore 0) per verificare la presenza di un eventuale boot, ovvero dei caratteri *CBM* nei primi byte del settore predetto.

Il boot, è chiaro, viene comunque eseguito solo se il drive risulta idoneo a svolgere tale mansione.

Il modo migliore per apprendere queste cognizioni, non è comunque quello sperimentato dal nostro lettore...

QUESTIONE DI BANCO

Per quale motivo, mentre con il C/64 non sorgono problemi, col C/128 non riesco a modificare il colore dello schermo con un semplice...

Lda #00

Sta $D020

Come posso risolvere questa difficoltà?

(Enrico Ulivieri)

L'operazione citata funziona per il C/128 esattamente come per il C/64, e, per i non conoscitori del linguaggio macchina, corrisponde all'istruzione basic **Poke 53280, 0**.

L'unico problema, può essere rappresentato dalla configurazione di banco attiva.

La locazione **$D020**, infatti, deve essere intesa come "locazione D020 del banco 15", altrimenti la mini routine non può sortire alcun effetto.

Quindi, da un punto di vista pratico, le scelte possibili sono tre:

1) Allocare la routine in una sezione di Ram al di sotto dell'indirizzo $4000, assicurandosi prima dell'attivazione che sia attivo il banco 15. Per esempio..

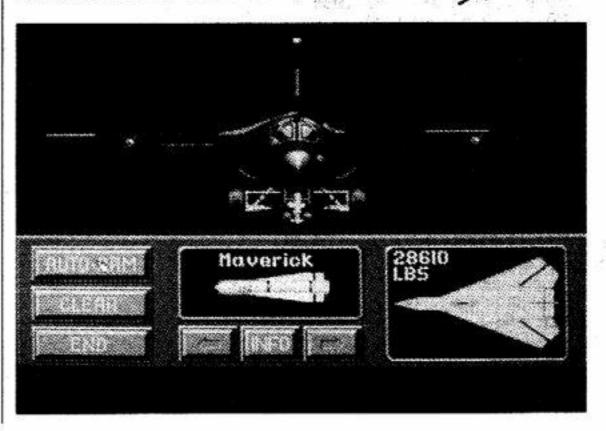
```
1300 LDA #00
1302 STA $D020
1305 RTS
```

... cui dovrà seguire, per l'attivazione da basic, Bank 15: Sys 4864 (l'istruzione Bank può essere omessa se si è certi di trovarsi già in banco 15).

La routine deve trovarsi al di sotto di $4000 per un motivo logico: se si trovasse, per esempio, in $C000 (49152), dovrebbe necessariamente risiedere in banco 0, considerando che in banco 15 a quella locazione corrisponde della memoria Rom (non accessibile in scrittura).

Una Sys 49152 in banco 15 non "troverebbe" dunque la nostra routine, ma del codice ignoto.

D'altra parte, qualora si settasse il banco 0 per raggiungere le istruzioni assembly da noi inserite in $C000, non sarebbe più accessibile diretta-

mente la locazione $D020 di banco 15.

Per questa eventualità, si veda il punto 3.

2) Sistemare la routine in una porzione di Ram che può accedere direttamente a qualunque configurazione di banco semplicemente modificando il contenuto della locazione $FF00 (più volte esaminata proprio nell'ambito di questa rubrica).

Per default, quest'area è rappresentata dal Kilobyte più basso della memoria, ovvero da $0 a $400 (1024).

Tutto sta a trovare lo spazio necessario, senza interferire con le molte aree di vitale importanza per il sistema.

Se non si dispone di una più che ottima conoscenza della macchina, è meglio evitare questa alternativa.

3) Se si vuole essere certi del funzionamento di una routine quale che sia la configurazione attiva, piuttosto che un semplice STA $D020 è opportuno ricorrere alle routine di sistema **Indsta** oppure **Stash**, che "Pokano" il valore desiderato nella locazione di qualsiasi banco.

La prima, locata in $F7DA di banco 15, presenta appunto l'inconveniente di dover prima accedere a questo banco, mentre la seconda è decisamente più comoda, essendo richiamabile da qualsiasi banco per la sua allocazione in area comune.

L'utilizzazione di quest'ultima, rigorosamente in linguaggio macchina, prevede che, prima di un salto (Jsr) all'indirizzo $2AF (687), vengano settati i seguenti parametri:

In due **loc**azioni contigue di pagina zero (le prime 255) va posto l'indirizzo della cella di memoria ove si intende "storare" il byte desiderato (naturalmente, in formato basso / alto). Il numero della prima delle due locazioni appena considerate va immesso in **$2B9**.

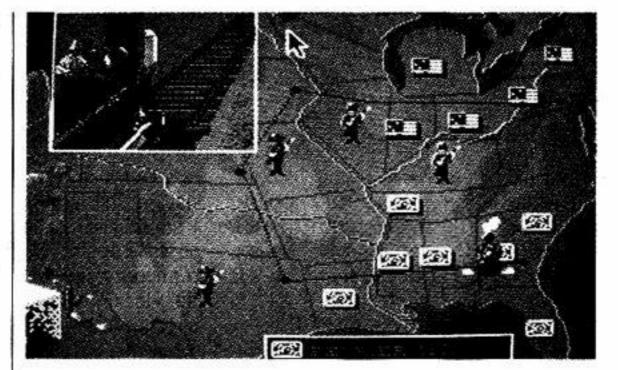
In **Acc**umulatore il byte da trasferire.

In **X** il banco desiderato, che però (ahimè) non è espresso in modo diretto (0 per banco 0, 15 per banco 15, eccetera), ma secondo particolari criteri seguiti dal sistema. Per approfondire la cosa, si vedano il n. 69 (pagina VIII di Campus) e n. 67 (riquadro a pag. 24) della rivista, oppure ci si limiti ad accettare in fede che 0 imposta il banco 15, $3F il banco 0, e $7F il banco 1.

In **Y** un eventuale offset, altrimenti va azzerato.

Si badi che quanto esposto vale solo per la routine *Stash*.

Indsta ($FDA7) va eventualmente adoperata allo stesso modo, ma utilizzando per il registro X il reale valore di banco (0, 15, 1, eccetera).

Tornando alla semplice routine del nostro lettore, se questa dovesse (per esempio.) essere allocata a patire da $C000 di banco 0, ecco che per svolgere il suo compito dovrebbe essere (per esempio.) configurata come nel disassemblalto di queste pagine.

Per verificarne il funzionamento, si impartisca (da Basic) **Bank0: Sys 49152.**

E' tutto, ma, se non si ha molta dimestichezza con l'Assembly, la scelta più facile risulta come ovvio la prima, pur dovendosi adattare a qualche limitazione.

```
C000   LDX   #$20    Byte basso e
C002   LDA   #$D0    byte alto di
C004   STX   $FB     $D020 in loc.
C006   STA   $FC     di pagina 0.
C008   LDA   #$FB    Segnala prima
C00D   LDA   #$00    Byte da poka re.
C00F   TAX           Banco 15 (X=0).
C010   TAY           No offset (Y=0).
C011   JSR   $02AF   Routine Stash.
C014   RTS           Return.
```

TRE CONSIGLI TRE

Posseggo un C/128 con solo il registratore, e vorrei sapere:
1) Mi conviene acquistare il drive, o aspettare di poter comprare un Amiga?
2) Procurandomi il drive, posso poi tenerlo per adoperarlo con Amiga?
3) Passando ad Amiga, vorrei poter adoperare programmi come il vostro Turbo Pascal; è possibile tramite il C/64 Emulator?
(Eugenio Tacchini - Rivergaro)

1) Pensa ad Amiga, e senza aspettare troppo.

2) Scordatelo.

3) Teoricamente possibile, praticamente da non prendere in considerazione. Amiga ha i "suoi" compilatori Pascal.

DA SCHERMO A STAMPANTE

Esiste un comando per stampare la grafica realizzata con le istruzioni Box, Circle, Draw, eccetera? Nè il libretto di istruzioni della mia stampante, nè quello del computer ne fanno cenno.
(Federico Rossitto - Chieri)

No, nel C/128 (come pure nel C/64) non esiste una routine interna che produca quella che viene definita una **Hard Copy** dello schermo, ovvero una sua copia integrale su carta tramite la stampante. Il che, però, non significa che è impossibile, ma solo che è necessario provvedere da sè.

Specificamente per il C/128, sulla nostra rivista sono già state pubblicate due routine che assolvono a questo compito: una per lo schermo in bassa risoluzione (n. 53 pag. 100) ed una per la grafica hi-res, quella cui fa riferimento il lettore (CCC n. 52 pag. 39).

Riproporle in questa sede sarebbe impossibile, per cui non resta che invitare gli interessati a rivolgersi al nostro servizio arretrati.

RICHIESTE E PROTESTE

Perchè non spiegate come si programma il tasto Alt, che sul C/128 è del tutto inutilizzato? Avrei, inoltre, alcune obiezioni da muovere alla vostra rivista: i dischetti che accludete ogni tanto non sono una scusa per far pagare il giornale 6000 lire? Pensate che quello accluso al n. 72 era completamente vuoto, neanche formattato! E poi, fareste bene a continuare a trattare il C/128, altrimenti metterete l'utente di fronte a due alternative: cambiare computer, oppure cambiare rivista.

(un 128ista disperato)

Prima di passare ai toni forti, liberiamoci subito della richiesta tecnica con un classico "già fatto".

La programmazione del tasto **Alt**, nonchè di tutta la tastiera, è stata affrontata solo pochi mesi fa, ed in modo molto esauriente (immodestia a parte), sul numero 67 della rivista.

Cosa che già consente un avvio alla polemica impostata dal "disperato" lettore, evidentemente non molto attento nonostante poi stigmatizzi la mancanza di pubblicazioni inerenti il C/128.

Quanto al dischetto, se difettoso, non c'è problema: basta inviarlo al nostro servizio arretrati, e verrà sostituito prontamente. Continuare a trattare il C/128, è invece un capitolo in fase di chiusura.

La decisione, peraltro abbondantemente ritardata, è resa ormai obbligatoria non già da una nostra scelta arbitraria, quanto piuttosto dalla politica commerciale della Commodore.

Che ha sempre abbandonato al suo destino lo sviluppo di questo modello, sopravvissuto forse più per il C/64 che si porta a corredo, togliendolo infine dalla produzione. Forse anche con un po' di ingiustizia, ma, d'altra parte, non sarebbe assurdo se le pagine della nostra rivista fossero ancora occupate da articoli sul Vic 20 o addirittura sul Pet (quanti di voi ne conoscono l'esistenza?), togliendo spazio alla fascia di utenza maggioritaria, ovvero quella rivolta ad Amiga e C/64?

Quanto all'alternativa / ricatto posta dal lettore, è quantomeno poco realistica.

Certo, si può anche cambiare rivista, ma se si parla di C/128, dove sta l'alternativa?

Chi altri ha mantenuto (e manterrà) fino all'estremo una rubrica di Posta (per non parlare degli articoli) dedicata solo a questo computer?

Non rimane dunque che pensare alla seconda alternativa (magari tenendo per ricordo il buon vecchio otto bit, che contiene pur sempre un C/64 ben vivo e vegeto!), a meno che non si sia già talmente padroni del proprio 128 da considerarsi autosufficienti.

Non va dimenticato, poi, che uno scambio (anche tecnico) di notizie tra gli utenti può sempre continuare tramite la nostra **BBS** e la sua area messaggi, fino alla triste dipartita dell'ultimo esemplare.

Con ciò non si pensi che la cosa ci renda felici, tutt'altro, ma... c'est la vie.

QUALE SOFTWARE

Ho acquistato un C/128 d'occasione, e vorrei adoperarlo più che altro per archiviazioni e videoscrittura, anche se il computer è "vecchio" (in fondo l'ho pagato poco...). Che cosa mi consigliate?

(Roberto Andreon - Cavallino)

Per questi usi, anche senza ricorrere alla smisurata biblioteca di software per C/64, sono facilmente reperibili due programmi di ottimo livello: il **Superbase128** ed il **Superscript128**, che sfruttano fino in fondo le capacità del computer, migliorando di parecchio le già ottime prestazioni degli omonimi pacchetti software riservati al C/64.

DA COMPUTER A COMPUTER

Uso il mio C/128D principalmente per elaborare (con Superscript) dei testi, che ho la necessità di trasferire in ambiente MS-Dos. E' possibile collegare il 128 al Commodore 30 III, che ho in ufficio, senza ricorrere a modem?

Faccio presente di essere un semplice utilizzatore dei computer, non un esperto di informatica.

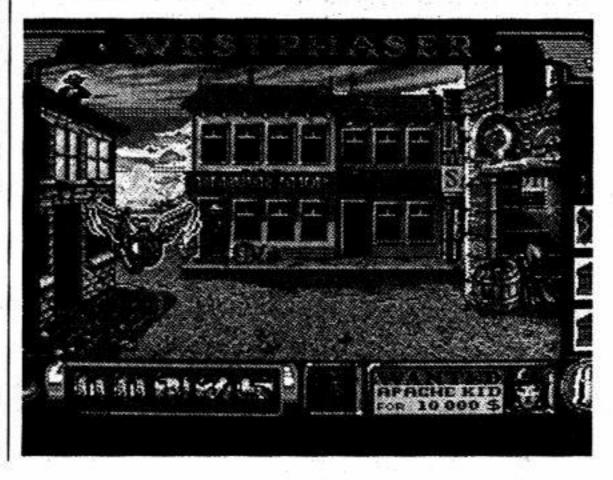
(Giuseppe Vavassori - Lucignano)

Sì, si può fare. Per effettuare il collegamento, è necessario anzitutto dotare il C/128 di una interfaccia seriale **RS232**, reperibile nei negozi specializzati o rivolgendosi ad alcune ditte che effettuano vendite per corrispondenza.

L'interfaccia, assolutamente uguale per C/64 e C/128, si connette alla porta utente sul retro del computer, e adatta i segnali elettrici del 64/128 allo standard delle macchine Ms-Dos (ma anche Amiga).

Occorre poi acquistare un cavo seriale, di quelli che comunemente servono per i collegamenti ai modem, al quale va apportata una modifica: bisogna invertire, all'interno di uno dei due connettori posti alle estremità del cavo, i collegamenti esisteni tra i piedini 2 e 3.

La numerazione, di solito, è stampigliata sul supporto in plastica dei piedini stessi.

Se non si è in grado di svolgere l'operazione (una semplice saldatura), ci si può sempre rivolgere ad un elettrotecnico oppure acquistare il cavo specifico, precisando

al rivenditore (esperto...) lo scopo desiderato. Il cavo va quindi collegato alla porta seriale del PC ed all'interfaccia RS232 del C/128. A questo punto non resta che affidare la trasmissione dei files Ascii a due programmi di comunicazione, che devono "girare" in contemporanea sui due computer. Un problema cui badare è la differenza esistente tra gli standard Ascii del C/128 (o C/64) e del PC (o Amiga), per cui i files non possono essere trasferiti così come sono, o risulterebbero in ogni caso inutilizzabili. La soluzione più efficace sarebbe quella di usare dei programmi appositi per il trasferimento, o per una conversione preventiva dei files (per esempio. quelli pubblicati all'uopo sul n. 49 della rivista), ma si può anche ricorrere, sul C/128 in modo 64, al programma **Vip Terminal**.

Questo, infatti, consente di settare la trasmissione di files Ascii operando automaticamente le necessarie modifiche al testo per adeguare lo standard Cbm all'Ascii dei PC (opzione **Cbm Conversion** dal menu **Files**).

Occorre, in questo caso, inviare i testi con l'opzione **Transmit Disk File** del Vip-Terminal, predisponendo il programma adoperato sul Pc in modo **Capture, Ascii Receive**, o qualcosa del genere (dipende dal software di comunicazione usato).

Se, invece, i files del C/128 sono preventivamente trattati per compensarne le differenze, allora è anche possibile adoperare protocolli di trasmissione più veloci, come per esempio, l'**Xmodem**.

In ogni caso, pur non ricorrendo all'uso di modem, e pur non essendo indispensabili specifiche cognizioni informatiche, sarà necessario impratichirsi con il software di comunicazione.

DUBBI ASSEMBLY

Potreste dirmi da che indirizzo partono i dati Ascii che le Rom routine del basic Fout ($AF06) e Val1 ($AF09) generano? Inoltre, non ho ben capito una cosa: il C/128 attiva gli IRQ in base al raster, e ad ogni interrupt viene messo in $D012 un valore. Come posso io inserire un mio valore senza che il sistema operativo ne metta un altro?

(Mirko Panciri - Padova)

Procediamo con ordine, dissipando anche le perplessità dei non addetti ai lavori (Assembly, si intende).

Le due routine cui fa cenno il lettore, riguardano entrambe il trattamento della cosiddetta notazione in virgola mobile, da noi esaminato più volte anche a proposito del C/128 (vedi ampia tabella sul n. 70 a pag.14).

Per lo più, queste routines servono ad effettuare calcoli matemetici facendoli svolgere direttamente al computer, soluzione decisamente più agevole che progettare in proprio complicate routines in linguaggio macchina.

Servono anche, ed è questo il caso in questione, a rappresentare in formato Ascii un certo valore numerico, ovvero a consentire la visualizzazione del risultato di una operazione matematica sullo schermo. Anche in questo caso, sono coinvolti dal sistema i due cosiddetti Accumulatori in virgola mobile (**Fac1** e **Fac2**), manipolabili come appunto descritto nell'articolo del n. 70.

La routine **Fout ($AF06)**, in particolare, crea una stringa di caratteri che rappresenta il valore numerico contenuto in FAC1, mentre **Val1 ($AF09)** svolge il compito opposto: preleva una stringa, e la trasforma nel corrispondente valore Floating Point (in virgola mobile).

Fout opera sul contenuto di Fac1, e la corrispondente stringa è depositata nelle locazioni da $100 in poi, con un byte zero che ne segnala la fine. Inoltre, come primo carattere conterrà uno spazio vuoto se il numero era positivo, oppure il segno meno.

Per attivare Val1, invece, è necesario depositare la stringa in Bank1 (sempre che non ci sia già), e in $24-$25 (dec 36 - 37) l'indirizzo ove essa è allocata (formato basso / alto).

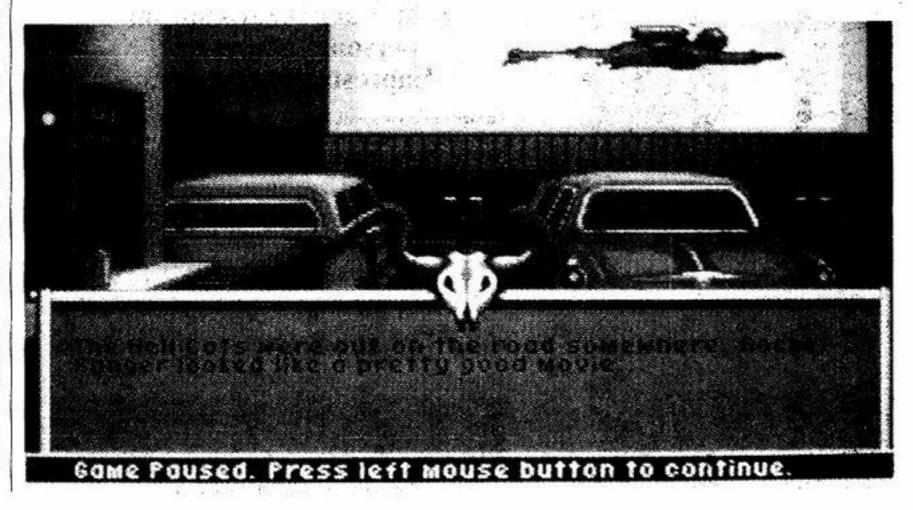
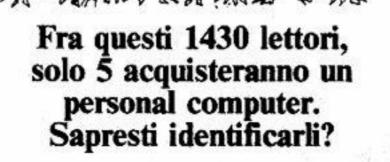
Il valore corrispondente, sarà depositato in Fac1, ed il sistema si troverà configurato, dopo l'esecuzione della routine Val1, in banco 14.

Non si dimentichi che la manipolazione di valori in virgola mobile, nonché dei due Fac, è da affidare ad apposite routines del sistema operativo, anch'esse descritte nel già citato articolo.

Quanto al registro $D012 del Vic, il cossiddetto **Raster Compare Register**, la risposta al quesito è molto semplice. Quando si accede in scrittura a questo registro, in realtà non si "scrive" nulla in esso, o almeno non in maniera definitiva.

Il valore eventualmente da noi immesso, viene memorizzato dal Vic in un suo particolare registro interno non accessibile in alcun modo, pronto per generare un interrupt (rilevabile con un esame del registro $D019) ogni qual volta il raster raggiunge la linea di scansione corrispondente.

Per chi volesse approfondire l'argomento, ecco pronta la solita "citazione": CCC n. 61, pag. XIV di Campus.

Fra questi 1430 lettori, solo 5 acquisteranno un personal computer. Sapresti identificarli?

La ragione per cui spesso si scelgono dei mass media anche per beni come un computer è semplice: nessuno sa dove è l'acquirente potenziale, nè come raggiungerlo.

I costi di questo sistema di contatto sono elevati e per abbassarli alcuni pianificatori preferiscono rivolgersi alle riviste specializzate. Ma anche in questo caso, il rischio di spendere male i soldi del cliente è elevato. La maggior parte delle riviste d'informatica professionale, infatti, va a coloro che il computer lo hanno già. Fare pubblicità su questi periodici è come reclamizzare frigoriferi in Alaska. Nel settore dell'informatica solo una pubblicazione professionale, Computer/Personal Computer, è letta da acquirenti potenziali di calcolatori. Naturalmente è distribuita in edicola e le sue vendite rispecchiano il numero di calcolatori che il mercato assorbe annualmente.

Il magazine mensile di "Computer"

VERMI, INSETTI, BACHEROZZI & VIRUS

Virus, chi li fa, che cosa sono, da dove vengono e come combatterli; una volta per tutte.

di Ascanio Orlandini

Tratteremo di un argomento che probabilmente terrorizza i neo-possessori di Amiga. Imparerete che cosa sono effettivamente questi programmi, veri "gioielli" di programmazione (più o meno diabolici), come lavorano e come ci si può difendere. Imparerete quindi a non temerli più ed a considerarli esattamente per quello che sono: programmi che seguono un algoritmo preciso, il più delle volte assai banale, e quindi assolutamente *non* intelligenti, come qualunque altro programma.

ALL'INIZIO

Il primo virus apparso pare che sia stato un programma, scritto da un ignoto americano, che si limitava a riprodursi all'infinito, ispirato alle piante rampicanti da cui ha preso il nome che, tradotto, era **Edera**. Girava sui primi sistemi Ms-DOS e non provocava danni. A questo prototipo se ne sono aggiunti altri nel corso del tempo, arrivando all'ormai famoso "virus della pallina", che disturba lo schermo con una pallina (una **'O'** piena) rimbalzante sullo schermo e a quello che provoca una cascata di lettere dall'alto.

Ai simpatici scherzetti se ne sono aggiunti altri, meno gradevoli, dagli effetti disastrosi, come quello che paralizzò, non tanto tempo fa, il Pentagono. Grande successo hanno avuto anche, sempre nel mondo Ms- DOS, i cosiddetti programmi **Trojan** che, come il mitico Cavallo di Troia, mentre dichiarano esplicitamente di svolgere qualche interessante compito, in effetti formattano l'hard-disk o compiono simili diavolerie.

Tornando al nostro Amiga, il primo virus comparso sulla sua strada è lo **Sca-virus**, il quale, come "Edera", è stato il "pioniere": non faceva nulla di male, a parte il riprodursi e il manifestarsi dopo 7 reset Ctrl- Amiga- Amiga.

Lo *Sca*, quindi, ha dato il via al primo ceppo di virus amighiani, quelli cioè che risiedono nel boot-block. Vediamo di chiarire il concetto.

IL BOOT - BLOCK

La prima operazione compiuta dall'Amiga all'inserimento di un dischetto è proprio quella di leggere **due tracce** (la prima di ogni lato) del dischetto, lunghe **1024** byte ciascuna (chiamate appunto **Boot-Block**) e caricarsele in memoria. Le due tracce possono essere occupata dal **caricatore standard** per l'AmigaDos, da caricatori **speciali** come quelli dei *giochi protetti* oppure, appunto, dai **virus** che, sfruttando la peculiarità dell'Amiga-DOS, vengono caricati rapidamente *in memoria* e qui restano fino allo spegnimento completo della macchina, resistendo ad ogni tentativo di reset impartito dai soliti tre tasti. Sviluppo successivo si è avuto con il virus **Byte Bandit** il quale, oltre a duplicarsi su *tutti* i dischi non protetti da scrittura, **inchioda** la macchina dopo cinque minuti di attività. Questo virus dimostra l'alto livello di programmazione del suo autore: in 1024 byte di Linguaggio Macchina puro contiene, oltre al codice della riproduzione altamente sofisticato (che si "fonde" con la routine di sistema atta ad identificare **ogni** disco inserito), il codice per inchiodare il sistema e un contatore che numera le copie di virus che infestano gli altri dischi.

Come se non bastasse, è possibile riprendere il lavoro normale dopo l'interruzione mediante la pressione contemporanea dell'ultima fila di tasti (Alt+ Amiga+ Spazio+ Amiga+ Alt).

Come avranno fatto a sistemare tutto in un solo Kbyte?Il virus non si limita, quindi, a disturbarci durante il lavoro. Naturalmente, se si sovrascrive su un

boot-block occupato dal caricatore particolare di un gioco... addio programma!

IL DIAVOLO ESISTE

La situazione cambia drasticamente con la comparsa del **Lamer Exterminator**: virus diabolico, di cui sono comparse ben cinque versioni (**I, II, III, IV** e **Revenge of Lamer Exterminator**, di cui parleremo più avanti), che si propone di sterminare i **Lamer**, cioè quelle persone che non "cracckano" (sproteggono i programmi originali) e non programmano, ma si limitano a collezionare indiscriminatamente software e (magari) a rivenderlo. Il virus ha la peculiarità di caricarsi in memoria in una posizione casuale e di *crittografarsi* nel boot-block secondo una **chiave** diversa per ogni copia, per cui ognuna sarà diversa dalle precedenti, provocando una evidente difficoltà di individuazione. Oltre a ricopiarsi di boot in boot, dopo due reset (o dopo che siano stati infettati i boot-blocks di tre dischi), provvede a riempire, con la scritta **"Lamer!"**, la traccia random (casuale) di un dischetto sprotetto inserito.

Così facendo il malcapitato file andrà irrimediabilmente in **Read Error** e sarà perso per sempre. Il virus colpisce anche coloro che promuovono l'Amiga per un uso serio: difficilmente un database o un foglio elettronico viene messo in autoboot, ma non per questo è esente dal pericolo di essere colpito da un virus!

Quando nel caricare un programma capita di incontrare un requester di read-error che non sia stato preavvisato dai due agghiaccianti stridii che emette il drive quando tenta di leggere una traccia che porta un errore fisico, potete scommetere che si tratta dello zampino del lamer che ha corrotto i dati di una traccia senza aggiornarne il checksum per avere, appunto, questo "dolce" e puntatore osceno pochi minuti dopo un reset.

Esiste poi il **Graffiti** che è il primo e, per ora, l'unico virus che si presenta con una grafica rotante 3D e l'**Australian Parasite** che forse è il più simpatico ed innocuo: *sembra* che non si riproduca sui boot-blocks non standard (per cui giochi ed altri boot particolari sono salvi). Dopo la lettura di circa 600 blocchi di disco, infatti, **ribalta** lo schermo, che può tuttavia essere ancora usato!

Infine va menzionata una particolare versione di **Diskdoctor** che si "mangia" 10K di memoria dopo ogni reset e continua ad allocarsi in tre posizioni della memoria. Ha dato molto filo da torcere ai programmatori anti-virus: se lo si cancella da una locazione, poco dopo torna a posizionarsi!

Finita la veloce carrellata sui "Boot-Block virus" dell'Amiga, è giunta l'ora di esaminare il "nuovo ceppo" di batteri: quelli che si fondono con files eseguibili.

"FUSIONI" BOLLENTI

Il pioniere in questa categoria di virus è stato l'**Irq Virus** che, come i suoi predecessori (Edera e SCA) non fa nulla di male tranne il fatto di esistere e riprodursi. L'algoritmo (cioè la procedura) con cui agisce è molto singolare e degna di essere conosciuta...

con il programma stesso; se, invece, non riesce nel suo *primo* intento (file non eseguibile, manca la Startup-Sequence, ecc.) tenta di aggredire il comando **Dir** cercandolo sia nella directory **C:** che in quella **corrente**.

Quando si fa ripartire il sistema con un dischetto infettato dall'IRQ, verrà cambiato il titolo iniziale della finestra dell'AmigaDOS in **"Amigados Presents: The Irq Virus V 41.0"** e cercherà di ricopiarsi su tutti i dischi inseriti. La vera astuzia si nota quando viene inserito un dischetto con la protezione contro la scrittura inserita: il virus far apparire un requester **"Volume *nome* is write protect"** incitando l'(ingenuo) utente a sproteggere il dischetto.

Il "file-virus" **BGS-9** lavora in modo simile: legge la Startup-Sequence e mira al primo file chiamato; solo che, invece di fondersi con esso, lo copia nella directory **Devs:** col nome di **Spazio** (rendendolo invisibile), e si riproduce al suo posto. Eseguendo la Startup-Sequence, quindi, verrà eseguito il virus che, a sua volta, manderà in esecuzione il *vero* file (lo *Spazio* nella *Devs:*) in modo che la procedura appaia "normale". Similmente agisce anche il **Revenge Of Lamer Exterminator**, l'ultimo tentativo di sterminio dei lamer.

L'ULTIMO ASSASSINO

Il più recente virus non-boot-block è lo **Xeno Virus**, apparso recentemente oltre Oceano, che si propaga come il fuoco di paglia fondendosi con **Tutti** i fles caricati o lanciati.

Gli ultimi quattro virus sono sicuramente i più pericolosi, se non per quello che possono fare, per gli sviluppi futuri che sicuramente avranno.

Se, infatti, la limitazione ad un Kbytes ...

creare problemi agli utenti telematici che, se esenti dai boot-block, corrono il rischio di *downloadarsi* (prelevarsi dalle bbs) files e programmi con "aggrappato" l'indesiderato ospite.

POVERO E INDIFESO

Fortunatamente ci si può difendere con i cosidetti **Anti-Virus**, programmi che hanno, come *missione*, quella di liberare l'Universo dai virus informatici.

Ci troviamo, però, di fronte ad una situazione non troppo limpida che crea non poche ambiguità: basta vedere quanti programmi si raggruppano sotto la categoria "Anti-Virus".

Delle decine, per non esagerare, di tali programmi, oggi quelli validi si possono contare sulle dita di una mano. Se poi qualcuno osa cercare un antivirus che colpisca *tutti* i virus sia da boot-block che da file, sia in memoria che su disco, ha una sola possibilità: **Virusx V4.0**

La vera forza di questo antivirus dall'aspetto semplice e modesto è da ricercare nell'esperienza del suo autore (**Steve Tibbett**) che da anni si occupa della difesa virale di Amiga mediante un continuo aggiornamento del prodotto che, oggi, sa riconoscere ed eliminare, sia sul disco che nella memoria, qualunque nuovo tipo di virus che appare, sia da boot-block che da file.

Una grande svolta di livello qualitativo gli è stata impressa con l'ultima versione 4.0, prelevabile da bbs d'oltre Oceano, e distribuita dall'autore del presente articolo ad alcune Bbs del Nord Italia. Il programma, ovviamente di Pubblico Dominio e, come dice Steve stesso, **da distribuire più che si può, è** in grado di riconoscere, per la cronaca, dai classici SCA, Byte Bandit 1 e 2, Byte Warrior, fino ai più recenti e dannosi Revenge, Old e New Northstar (a detta di molti un AntiVirus...), Obelisk Softworks Crew, Pentagon Circle, Lamer Exterminator 1, 2 e 3, Graffiti, Disk Doctor, Australian Parasite; dell'altra specie riconosce invece l'IRQ, il Revenge of Lamer Exterminator, il BSG-9 ed il nuovissimo Xeno.

La confezione originale del programma (in origine un archivio **Arc**, ma si può trovare nei formati più disparati) è molto ricca dal momento che contiene, oltre al programma **VirusX**, un secondo programma chiamato **KV** (Kill Virus) di nuova versione 2.1 che permette di controllare la salubrità dei singoli file.

Mentre, infatti, VirusX controlla solo i programmi "sospetti" (primo della startup, ecc.), tramite KV possiamo controllare, per esempio, i programmi appena prelevati da una bbs o i files di un classico disco zeppo di utilities di provenienza sospetta per essere certi di non infestare il nostro computer.

Dulcis in fundo... per tutti i menzionati programmi eseguibili (VirusX e KV), oltre alle **Documentazioni** di rigore (ovviamente in inglese), vengono forniti **tutti i sorgenti in linguaggio C Lattice!** Per chi ne mastica un pochino, è sicuramente cosa degna di nota, che garantisce il prodotto: il programmatore stesso invita ad esaminarlo e di compilarlo se avessimo dubbi che si tratti di un *trojan*.

Naturalmente non è sufficiente possedere gli antivirus per essere tranquilli: bisogna procedere ad una sistematica disinfezione dell'**intera** collezione di dischetti; in seguito è "sufficiente" un rigoroso controllo di ogni dischetto che si avvicina all'Amiga.

Se presterete la necessaria attenzione, e riuscirete a non farvi scappare nessuna "new entry" resistendo alla tentazione di provare subito i nuovi dischi, vivrete felici e contenti con la vostra sana Amiga.

CONSIGLI

Un paio di raccomandazioni e consigli per concludere. Quando VirusX segnala di avere incontrato un Boot-Block non standard (**Not A Standard Boot-Block** oppure **Unknown Boot- Block**), *non vuole assolutamente dire che e' un virus!* Vuole semplicemente informare che ha letto un boot-block che non corrisponde al caricatore standard AmigaDOS e che non viene riconosciuto tra i virus a lui noti. Le logiche conclusioni sono due: o è un nuovo virus, oppure è il caricatore di un gioco o una presentazione.

Grazie alla recente data di emissione della versione 4.0, la prima ipotesi è da scartare; si raccomanda quindi di *non installare mai* un simile boot-block. Pena, soprattutto per i giochi, il conseguente mancato funzionamento del software.

Sempre sottoponendo all'analisi un dischetto-gioco protetto, con conseguente sfasamento della struttura Dos, nel momento in cui l'antivirus controlla la startup-sequence e gli altri file a rischio, provocherà l'emissione di numerosi e fastidiosi requester che dovrete tranquillamente **cancellare**.

L'esame del bootblock avverrà poi correttamente; vi accorgerete che simili protezioni si accompagnano sempre da bootblock sconosciuti (unknown) in cui risiede il nuovo caricatore.

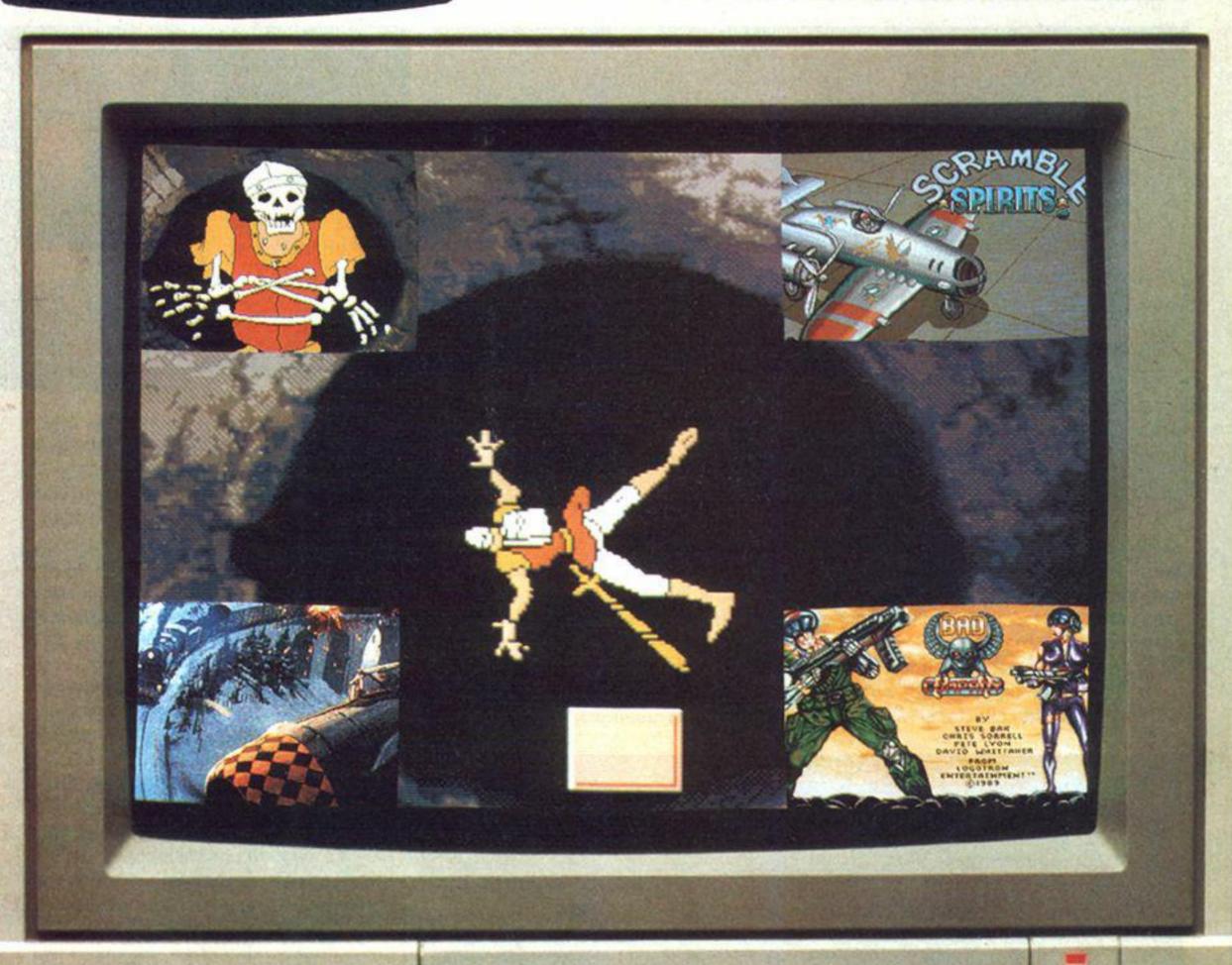
Un'ultima cosa: questo gioiellino di programma risente di violente compattazioni, per cui non comprimetelo con crunchers vari (quelli che creano arcobaleni, "terremoti" o che colorano il puntatore, per intenderci ...) perchè, pur guadagnando un po' di spazio, correreste il rischio di non fargli compiere bene il suo lavoro.

Naturalmente i programmi VirusX 4.0, KV 2.1 e relativi Docs e sorgenti C verranno inclusi nella prossima **Amigazzetta:** un motivo in più per non farsela scappare.

SCRAMBLE SPIRITS

AMIGA

Commodore MODEL 1081

POWER

SCRAMBLE SPIRITS

La nostra nazione è stata attaccata da una misteriosa torma di aeroplani, che, tanto per cambiare, scendono dall'alto verso il basso su di uno schermo a scorrimento orizzontale.
Lo scopo, come vogliono tutti i giochi "spara spara" (in inglese *shoot 'em' up*) è di sparare a tutto ciò che si muove.

Ad esempio anche ai carrarmati ed alle fortezze volanti che discendono dall'orizzonte aritificiale, rappresentato dal bordo superiore del monitor.

Si può giocare in uno o due avversari, a turno.

Sono previsti incentivi o bonus particolari, come ad esempio vite extra ogni tanti punti e cannoni potenziati, da raccogliere "al volo" sullo schermo, che aumentano la potenza di fuoco.

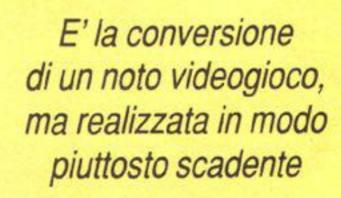
La tecnica

La grafica bidimensionale è quanto di peggio si possa offrire con Amiga. Lo scrolling, pur nella sua semplicità di fondali e nella lentezza, riesce tuttavia ad essere impreciso.

Gli sprite sarebbero di rifinitura pietosa anche su di un C/64 e le scelte dei colori sono del tipo ammazzapupille.

Il Voto

Un pessimo programma. 4.

E' la conversione di un noto videogioco, ma realizzata in modo piuttosto scadente

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Shoot 'em' up
Softhouse: Grandslam

Il tanto atteso(?) seguito di Dragon's Lair è finalmente giunto in Italia

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade 3D
Softhouse: Entertainment International

SINGE'S CASTLE

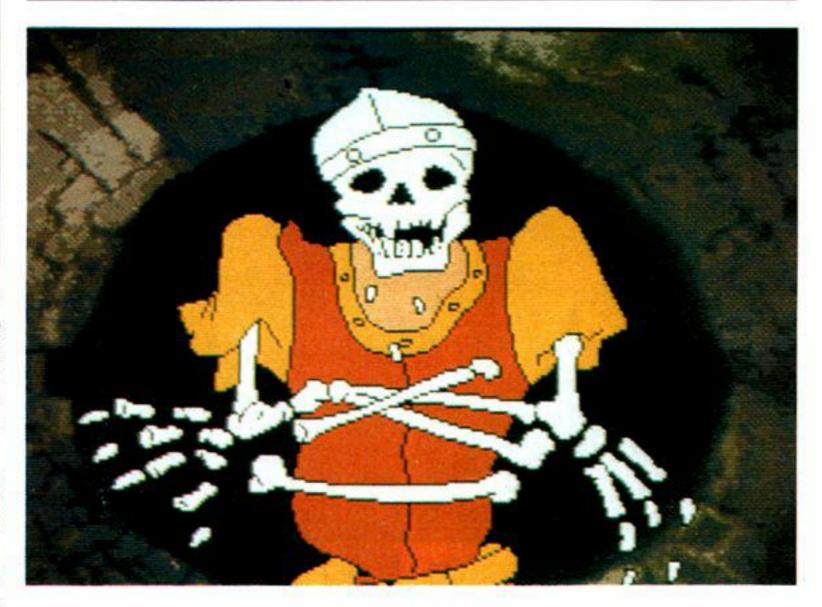
Don Bluth ha sfornato un altro dei suoi videogiochi, dopo *Dragon's Lair e Space Ace*. Come sempre, il programma presenta una grafica eccellente, sia per quanto riguarda gli scenari, sia per quanto riguarda l'animazione tridimensionale, veramente da cartone animato.

Anche l'interattività è al minimo: serve solo un colpo di joystick al momento giusto per superare la fase ed ammirare la sequenza di animazioni successiva.

La novità risiede nei tre livelli di difficoltà: il primo è semplice e "dura" soltanto otto fasi, con un box rettangolare che indica, al momento giusto, quando e come muovere la manopola del joystick per superare la fase. In tal modo, ad un giocatore di levatura media, serve probabilmente soltanto mezz'ora per completare tutte le sequenze a disposizione.

Nel secondo livello non si ha la freccia di aiuto, si ha meno tempo utile per fornire la risposta necessaria col joystick e non si può ovviamente accedere all'eccezionale sequenza finale. Questa è giocabile soltanto al livello superiore, che è composto, complessivamente, da venti fasi e richiede reazioni precise al fulmicotone.

La Tecnica

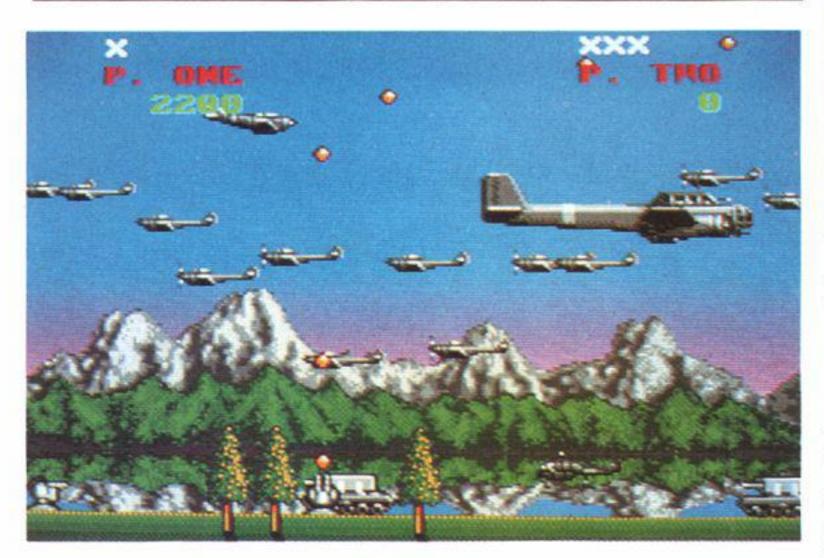
Se si possiede un Amiga 500/2000 con soli 512K non si perde una buona parte degli effetti grafici e sonori previsti dal gioco, che comunque continua pregevolmente a funzionare. E' compatibile con processori 68000/10/20/30, può salvare sino a cinque partite contemporaneamente, e può essere installato su disco rigido, sebbene da dischetto carichi già sino a 10 volte più velocemente rispetto alle precedenti produzioni.

Tutto ciò, oltre all'eccezionalità della grafica e degli effetti sonori, chiarisce quale sia la pregevolezza tecnica di questo programma.

Il Voto

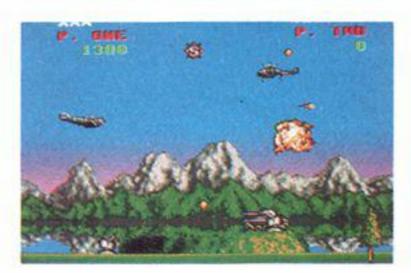
Più che un videogioco, un eccezionale "dimostrativo" grafico, che spreme ai limiti attuali le possibilità grafiche di Amiga. 7-.

P-47

Il P-47 Thunderbolt fu uno dei migliori caccia durante l'ultima guerra

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Shoot 'em 'up
Softhouse: Firebird

Il glorioso P47 ha dato il nome ad una conversione dell'omonimo videogioco della *Jaleco*, in cui il giocatore (eventualmente in gara con un compagno) siede nella carlinga di uno di questi velivoli per sparare a tutto quanto compare dal lato destro dello schermo.

P-47 appartiene alla categoria di giochi detti in gergo *Horizontal Side - Viewed Scrolling Zapper*.

Con il nostro aeroplanino (visto di profilo al centro dello schermo, spostabile ortogonalmente con il joystick) dobbiamo combattere contro nemici di terra e di aria che entrano dal lato destro del video. L'arma a nostra disposizione è un cannoncino frontale, che può essere affiancato da altre armi quando si colpiscono gli elicotteri nemici e si raccologono, in tempo utile, i simbolini che lasciano.

Inoltre sono disponibili bombe che vengono sganciate quando si spara con il pulsante di fuoco. Maggiore è il *livello di energia* guadagnato con l'aeroplano, maggiore è la sua potenza di fuoco, che non va persa nemmeno quando si perde una vita. A fine livello si trovano i consueti "guardiani", molto tosti, rappresentati da un treno, un aeroplano, un carrarmato, una nave e da altro ancora.

La tecnica

I colori sono molto buoni e gli sfondi semplici ma molto lunghi e decisamente rifiniti.

Gli effetti sonori sono di quelli "intensivi", cioè si sconsiglia di giocare P-47 dopo le dieci di sera a volume alto. Lo scrolling, data anche la semplicità della grafica, è preciso e fluido.

Il movimento ed il controllo sono decisamente buoni.

Il Voto

Un gioco semplice ma "onesto", ben rifinito. 7 1/2.

BAD COMPANY

Un videogioco che non può non ricordare Space Harrier: è identico!

Computer: Amiga inespanso
Gestione: Joystick o Mouse
Tipo: Arcade
Softhouse: Logotron

Si inizia con una schermata stile *Speedball*, dove si sceglie l'alieno che ci rappresenta sullo schermo: *Athena, Lance Gordon, Bruce North, Leroy Henrinksen, Stefan Hunter, Met Steinberg, Chronos Warchild* oppure *Sandi Kalishnov*. Non crediate, comunque, che la scelta cambi la strategia del gioco, che rimane essenzialmente uno "spara spara" con grafica prospettica 3D stile *Space Harrier;* la scelta soddisfa semplicemente l'occhio, che vuole anch'esso la sua parte (o no?). Il gioco si snoda su quattro livelli, con alieni, paesaggi e difficoltà ovviamente progressive.

In particolare, i mostriciattoli nemici sembrano più perfidi ed intelligenti di quelli di Space Harrier dal momento che si nascondono spesso molto bene dietro i muri per balzare fuori all'ultimo momento davanti al nostro *bazooka*.

Le armi hanno numerosi livelli di potenza e velocità di fuoco, per sopperire a varie esigenze, cioè sopprimere altrettanti tipi di nemici. Inizialmente se ne hanno due, selezionabili via barra spaziatrice, ma altre vengono fatte cadere dalla nave madre, che fornisce gentilmente, di tanto in tanto, i caricatori per gli scudi elettronici di protezione, che vengono intaccati dai colpi dei mostri.

La Tecnica

Gli sprites dei nemici sono molto variopinti e d'effetto: *meduse* saltellanti variopinte, specie di *gorillastri, funghi* ed *occhiacci* volanti con chiari problemi di congiuntivite sono soltanto qualche pallido esempio. Mancano rispetto a Space Harrier II i dragoni di fine livello, ma non se ne sente la mancanza.

Gli effetti sonori sono notevoli, anzi costituiscono, insieme alla grafica, la maggiore attrattiva del gioco, altrimenti piuttosto semplice per concetti ed idee.

Il movimento dello sfondo ha una buona inerzia ed è interessante la possibilità di utilizzare anche il mouse.

Il Voto

Peccato che sia un'idea non nuova. Buona tecnica. 6/7.

GALAXY FORCE

Il gioco dimostra che non sempre è possibile convertire con successo un "coin up"

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade spaziale
Softhouse: Activision

Non preoccupatevi, con il gioco non viene fornito il contorno della sala giochi: mostro ad otto zampe con monitor gigantesco ad alta risoluzione, seggiolino idraulico che trasferisce gli scossoni della guida direttamente ai vostri glutei, casse stereofoniche ad otto vie eccetera, eccetera...

IL GIOCO

Il *Quarto Impero* ha conquistato i cinque pianeti della nostra Galassia, collocando su ciascuno delle proprie basi. Scopo del gioco, di difficoltà progressiva, è di penetrare gli scudi difensivi di ciascun pianeta e farle saltare.

Si vola su di un'astronave leggera da guerra, armato di laser ad impulso e di schermi protettivi (tipo *Star Trek*) che hanno, ovviamente, una durata limitata nel tempo. La velocità è variabile agendo su appositi tasti di Amiga.

La prima fase della conquista di ogni pianeta si svolge quindi sull'orbita, con torme de caccia nemici che cercano di abbatterci e visione tridimensionale.

La seconda fase, migliore sotto vari aspetti, si svolge all'interno delle basi nemiche, dove bisogna imparare anche ad evitare i muri oltre ai velivoli nemici.

LA TECNICA

Il programma conserva ben poco del videogioco originale della sala bar, che utilizzava uno speciale microprocessore a 32 bit in grado di generare una dettagliatissima e velocissima grafica tridimensionale, accompagnato da un chip a 16 bit specializzato nella generazione di musica ed effetti sonori.

La grafica è "limitata" (si fa per dire) agli standard Amiga e non sfrutta che in parte la velocità di operazioni che consentirebbero i vari Blitter e Copper.

IL VOTO

Se si considera che il gioco originale non sarebbe comunque stato un granchè senza il monumentale supporto hardware citato prima (e che la conversione è scarsa) si ricaverà una chiara idea di che cosa pensiamo di questo programma. 5, non di più.

Ancora una simulazione sportiva? Si', ma con qualche novità degna di nota

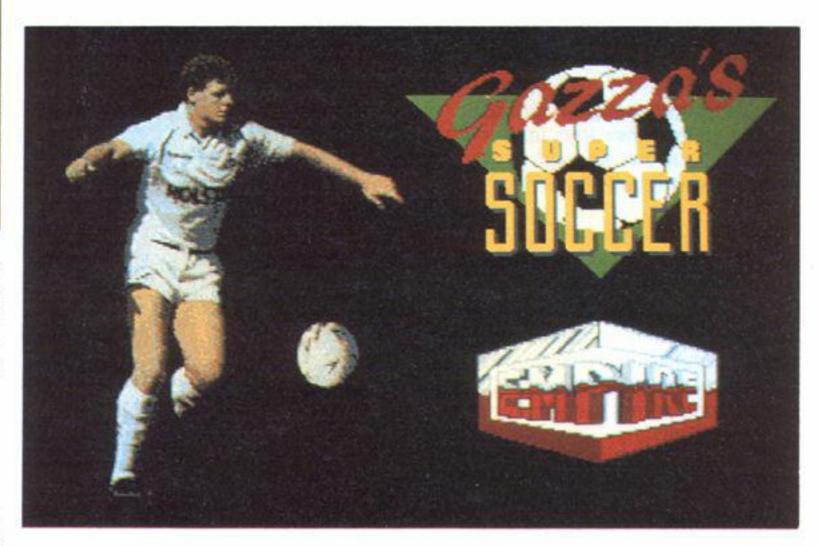
Computer: Amiga inespanso
Gestione: Joystick
Tipo: Calcio
Softhouse: Empirès

GAZZA'S SOCCER

Sebbene il gioco sia intitolato a un tale *Gazza*, è sponsorizzato dall'immagine di *Paul Gascoigne* che a noi, pur essendo cresciuti a pane e football, risulta tanto noto quanto lo può essere *Goethe* ad un analfabeta.

IL GIOCO

Incominciamo a pensare che la vicinanza di *Italia '90* rappresenti un ottimo stimolo non solo per dilapidare miliardi in faraoniche attrezzature sportive, ma anche per far sfornare alle softhouse una caterva di giochi basati su sprites più o meno rotondi ed omini in pantaloncini corti.

Si può scegliere tra quattro campionati eruopei: inglese, scozzese, francese ed italiano. Il regolamento è quello inglese, quindi per la vittoria fuori casa si guadagnano tre punti invece che due.

Ciascun giocatore è dotato di una propria personalità, differente da quella degli altri atleti: velocità, controllo di palla, propensione ai falli, colore dei capelli e della carnagione. Ciò contribuisce essenzialmente a variare la meccanica degli avvenimenti e ad introdurre una maggiore casualità nel gioco, aumentando il tempo di impratichimento necessario.

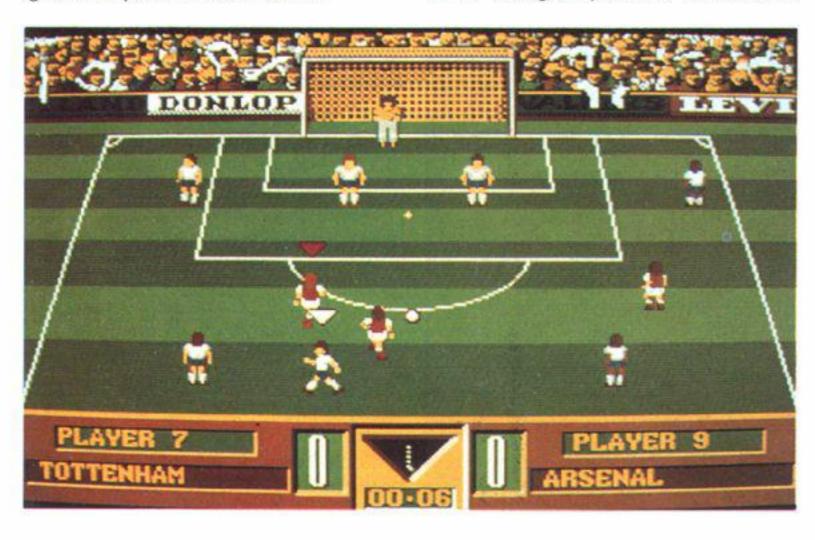
Il lancio del pallone è controllato tramite un triangolo posto in basso sullo schermo, che indica altezza ed effetto del tiro. Ciò serve a aumentare il controllo sul pallone, ma riduce la velocità operativa, che risulta quindi ben lontata da quella al fulmicotone di *Kick Off.*

LA TECNICA

Il campo di gioco è presentato lateralmente (quando il pallone è a centrocampo), dall'alto e da centrocampo (quando il pallone è in prossimità di una delle due reti). Ciò aumenta la giocabilità ed impone un rapido colpo d'occhio al viodegiocatore per controllare adeguatamente il pallone. Può quindi essere visto come un difetto seccante o come un tocco di originalità che aumenta il realismo del programma.

Gli sprites sono disegnati abbastanza bene, con un'animazione sufficientemenre fluida. Il pallone, una specie di croce, farebbe impazzire un fisico per la sua cinematica.

IL VOTO

Un gioco del calcio dotato di tocchi innovativi originali, ma che possono trasformarsi in *handicap.* 6 più.

A bordo di un otto volante bisogna sparare contro tutti; anche contro una nave!

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade rapido
Softhouse: Tynesoft

ROLLER COASTER

Simulare una corsa in montagna russa non sembra una buona idea, dato che difficilmente un joystick può trasmettere le sensazioni di stomaco in gola, eppure potrebbe essere interessante provare a simularla.

IL GIOCO

A parte la presentazione iniziale, tutta la visione del gioco è in prospettiva 3D semplificata dal posto di guida della vetturetta di una montagna russa, armata con un paio di bazooka.

Lo scopo è semplice: distruggere tutto quanto si incontra per conseguire punti.

Il gioco scorre in movimento continuo sulla pista dell'ottovolante con assai poca varietà, a parte gli oggetti incontrati, che variano da strane figure solide galleggianti nell'area, ad interi vascelli fantasma, probabilmente scappati dalla galleria degli spettri, che scaricano proiettili sul percorso.

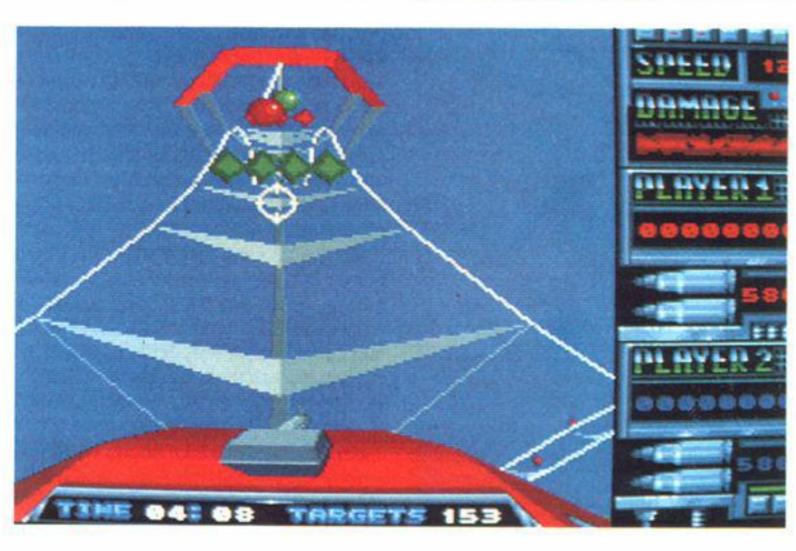
Durante la corsa contro la morte bisogna sparare per primi contro gli oggetti disseminati sulla pista, in quanto colpendone troppi si finisce col provocare il deragliamento della vettura.

LA TECNICA

Sebbene gli ingredienti per un buon gioco siano stati inseriti tutti, la grafica 3D è troppo traballante ed imprecisa per riuscire realistica e suggestiva.

Anche gli incentivi sono assai limitati, ed in generale è uno di quei programmi che, per concezione e tecnica di programmazione, sarebbe appena sufficiente su di una macchina ad otto bit, non certo su di un Amiga.

IL VOTO

Pochi incentivi, interazione assente, tecnica difettosa. 5.

SWORDS OF TWILIGHT

La leggendaria *ECA* deve buona parte della propria fama agli ottimi programmi prodotti in contemporanea con i primi vagiti di Amiga nel mondo informatico. Non sempre la sua partecipazione nel mondo dei videogiochi è stata felice, comunque, anche quando scendono in campo i programmatori del celebre *Archon*, come in questo caso.

L'idea era buona, le intenzioni anche. I risultati sono però deludenti

Computer: Amiga inespanso
Gestione: Joystick e/o tastiera
Tipo: Role Playing Arcade
Softhouse: Electronic Arts

IL GIOCO

La terra di Albione è sotto il potere degli *Shadow Lords*, i Signori delle Tenebre, che progettano di invadere anche le altre sette terre contigue per spargervi i propri maligni incantesimi.

Possono giocare sino a tre persone contemporaneamente, ciascuna delle quali controlla un personaggio scelto inizialmente da un set di una trentina, il cui scopo finale è di recuperare le *Spade del Crepuscolo*, unici strumenti per difendersi vittoriosamente dai nemici.

I personaggi si dividono in due categorie: *Maghi* e *Cavalieri*. I primi sono più abili nell'usare gli incantesimi, i secondi nei combattimenti puri.

Il gioco vero e proprio può essere svolto in due modi: seguire le richieste fatte dai regnanti e ritornare ogni volta a casa, oppure andare a zonzo cercando da soli le avventure da risolvere.

Lo schermo è suddiviso in cinque porzioni: la parte centrale visualizza il gruppo di avventurieri, seguendoli nel loro girovagare, con due possibili livelli di ingrandimento panoramico a seconda della fase, come nella serie di giochi *Ultima*.

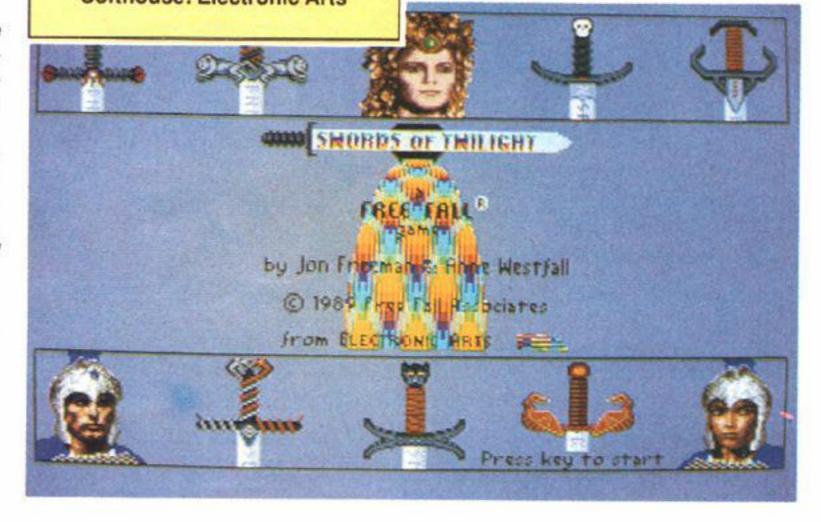
Le altre quattro porzioni indicano lo *stato* dei personaggi interessati all'azione, con eventuali stringhe di messaggio prodotte durante i dialoghi.

LA TECNICA

I suoni del gioco sono sgradevolissimi, tanto che è vivamente consigliato abbassare il volume del monitor durante le partite.

Gli effetti di supporto per i vari personaggi sono graficamente interessanti, ma tutto il resto è a poco più del livello consentito dal Basic del C/64, così come gli scrolling e le animazioni.

IL VOTO

Idea semplice, grafica scarsa, pochi incentivi e scarsa interattività. 5.

Ispirato a classici, come Thrust e Oids, è un game per un solo giocatore.

Computer: Amiga inespanso
Gestione: Joystick o Tastiera
Tipo: Arcade a labirinto
Softhouse: Arcana

ROTOR

L'ambientazione è in un futuro dove si può scegliere soltanto tra il lavorare in minere dell'*Antartide* oppure lavorare per l'esercito dei *RotoRaiders*. In questo caso il compito è di infiltrarsi nelle fortezze nemiche, distruggere le installazioni ostili e raccogliere gadget utili.

La nostra nave è vista dall'alto, in prospettiva piana bidimensionale, ed è armata con un cannoncino a fuoco frontale, un raggio trattore per raccogliere carichi, ed uno scudo energetico per proteggersi contro il fuoco nemico.

Si inizia con una sorta di esercitazione, dove si deve dimostrare di sapere sparare ed usare efficacemente il raggio trattore. Quando il responso è sufficiente, viene dato un codice di accesso (formato da tre lettere) che consente di iniziare le prime tre missioni. Queste consistono nel volare in fortezze nemiche, eliminare avversari a suon di laser e raccogliere *containers*, dove troviamo munizioni e carburante. Da raccogliere sono anche i cristalli energetici, convertibili in armi migliori o cariche energetiche extra.

La Tecnica

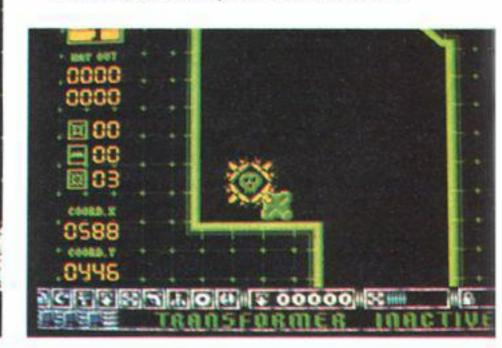
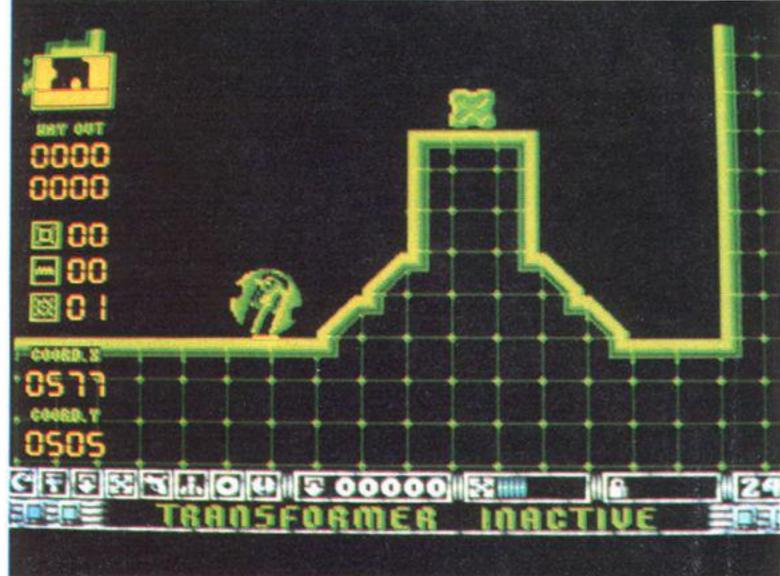
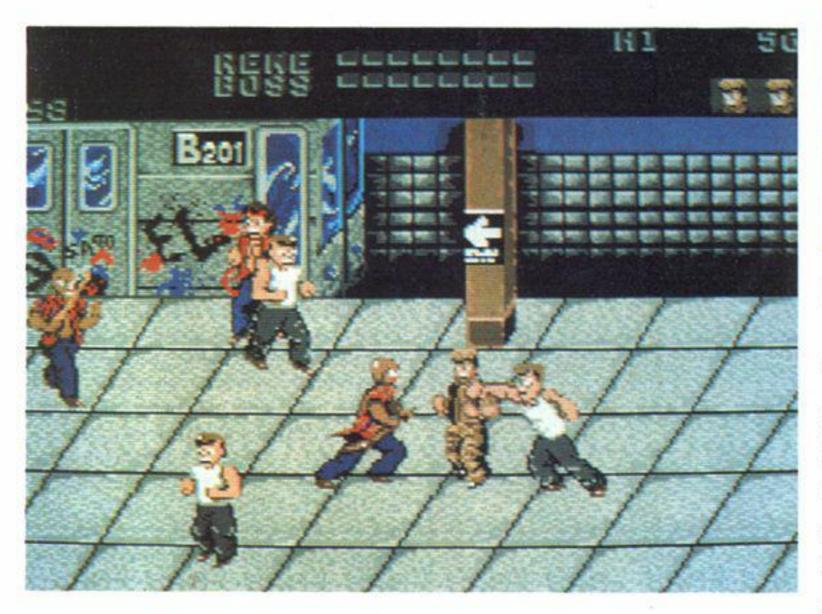
Nulla di notevole per quanto riguarda gli effetti sonori, che sono poco numerosi ma abbastanza buoni. La grafica è invece decisamente migliore, molto colorata e ben animata, sebbene certamente non sofiticatissima. Comunque il pregio del gioco sta negli incentivi, nelle difficoltà e nella caratteristica guida "inerziale" alla Meteoroids.

Il Voto

Interessante, semplice ma d'effetto. 7.

RENEGADE

Il protagonista è un bullo di periferia che trascorre la sua vita litigando con tutti

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade di combattimento
Softhouse: Taito

Un classico tra i videogiochi del bar, ora uscito nuovamente in versione "budget", cioè a prezzo bassissimo, per il nostro Amiga.

IL GIOCO

Si manovra uno sprite alto circa 40 pixel con la sagoma di un bullo di periferia il cui scopo, nella vita, non è quello di parcheggiare la moto "a pettine", ma di scazzotare con tutti i personaggi che incontra nei bassifondi della città.

Come esistono gli arcade spaziali, soprannominati dagli esperti *a fuoco totale* (perchè per procedere bisogna sparare in continuazione), un gioco come *Renegade* richiede semplicemente(!) di sparare pugni e calci senza soluzione di continuità.

In effetti è più difficile di quanto appena detto: bisogna imparare a schivare e colpire con appropriati movimenti del joystick i cattivoni che ci ostacolano secondo le armi che impugnano: catene, coltelli, pugni di ferro ed altri oggetti che ognuno di noi si ritrova abitualmente nelle tasche. La scenografia consiste in vari bassifondi di città passati in scorrimento continuo, realizzati con la stessa tecnica dei programmi da sala giochi, ma senza la medesima altissima risoluzione e dovizia di particolari.

LA TECNICA

Gli sprites non sono troppo vari e si muovono a scatti.

Anche il repertorio di colpi disponibili risulta piuttosto limitato. Tale particolare, se da una parte contribuisce a semplificare la vita del concorrente, dall'altra riduce la piacevolezza del gioco.

La grafica di scenografia è povera di dettagli, anche se fatta scorrere abbastanza bene.

Gli effetti sonori e la musica di sottofondo non sono molto entusiasmanti.

IL VOTO

Una conversione riuscita poco bene. Non si può andare oltre il 5 e mezzo.

TENNIS TABLE

Una bella partita di ping pong contro il computer, o contro un amico, è un'emozione conferita da tanti programmi per le vecchie macchine ad otto bit, ma certamente una versione per Amiga può essere molto più realistica.

IL GIOCO

Inizialmente si sceglie se giocare una regolare competizione od un training, senza punti. Quindi si seleziona, sempre con il mouse, se giocare la coppa del mondo od una gara singola, in uno o due giocatori, singolo o doppio. L'avversario viene scelto, a seconda del tipo di torneo, tra una ventina di nomi più o meno esotici, corrispondenti a giocatori più o meno bravi e con tecniche di gioco ovviamente differenti.

Notevole è la possibilità di scelta successiva, ovvero tra sei differenti tipi di racchetta ed impugnatura, a seconda del proprio stile personale. Infine viene chiesto il tipo di device di input (joystick o mouse) e la porta di assegnazione al/i giocatore/i, il numero di set da vincere,d il numero di punti per set (11 oppure 21).

Se sopravvivete a tutte queste regolazioni, che comunque consentono di personalizzare e variare il gioco a piacimento aumentando le possibilità di variazioni su di un tema altrimenti invariabilmente monotono, potete entrare nel gioco vero e proprio. Compare il tavolo verde scuro, con due file di tifosi ai bordi statici e le due racchette sospese con un torso di mano attaccate. Per colpire la palla basta premere il pulsante quando arriva alla distanza giusta. Gli effetti e la potenza al colpo vengono inferti col movimento della racchetta durante il contatto con la palla, ma richiedono parecchia pratica.

Due mani "armate" di racchette compaiono sullo schermo per disputare la partita

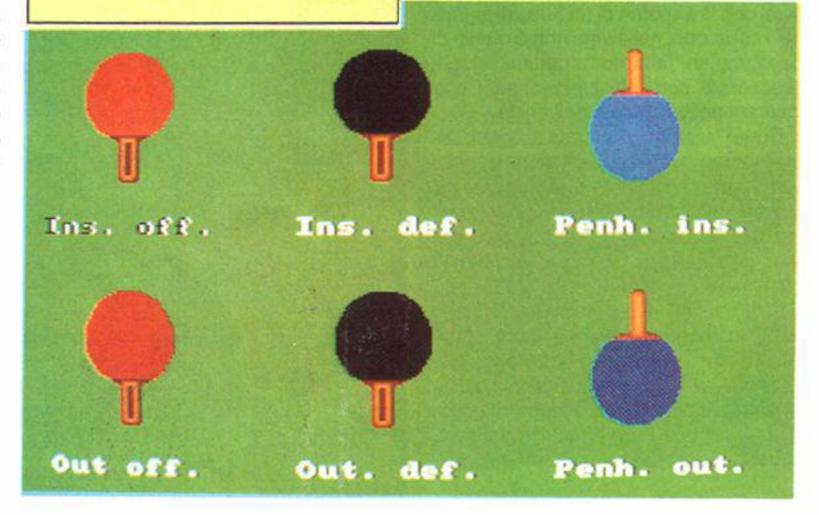
Computer: Amiga inespanso
Gestione: Mouse o Joystick
Tipo: Gioco di simulazione
Softhouse: Ormellysoft

LA TECNICA

Il gioco è estremamente semplice, con le racchette che si muovono rapidamente e fluidamente, ed una semplice animazione per simulare il movimento quando si porta il colpo.

Nessun preziosismo: spettatori fermi, effetti sonori ridotti al minimo ("tock" della palla sulla racchetta) e finestra standard che compare ad indicare la situazione di games e set dopo ogni punto. Alla fine si vede anche una brevissima animazione con delle tartarughe che giocano sul tavolino a lato e che suonano le prime note di BAD in un complessino sotto il tavolo.

IL VOTO

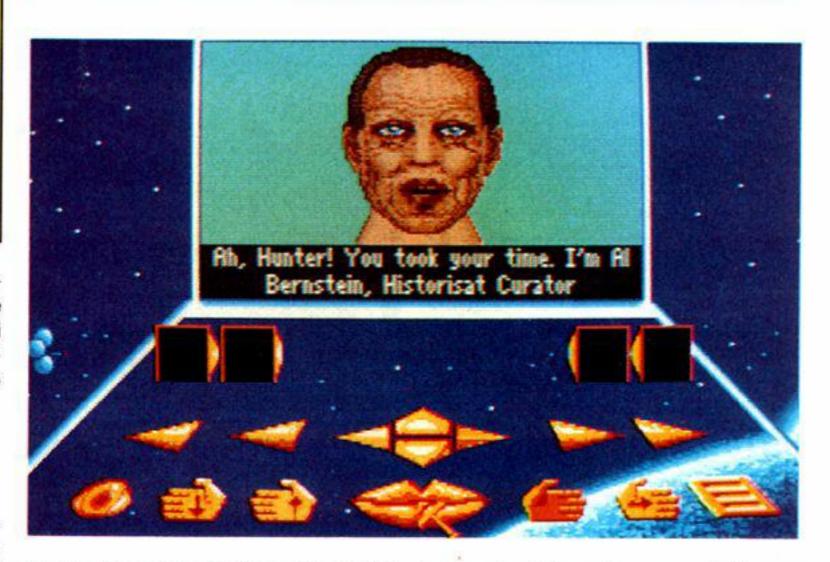
Un ping pong estremamente semplice, come è il gioco vero, ma piuttosto realistico, senza preziosismi ma con molte varianti. 7.

Viaggiare nel tempo ed interrogare personaggi famosi per evitare di soccombere

Computer: Amiga inespanso
Gestione: Mouse
Tipo: Avventura, grafica animata
Softhouse: Empire

TIME

Time è il tipico gioco che può interessare sia gli appassionati di avventure classiche (in particolare gli amanti di *Chrono Quest*), sia quelli di arcade, correndo magari il rischio di non soddisfare nessuno dei due.

IL GIOCO

Nell'anno 2047 siamo chiamati sul satellite *Historisat* in orbita intorno alla terra. Uno scienziato è riuscito a vedere il futuro, dove un androide chiamato *Mek* è sfuggito al controllo dei suoi creatori umani ed ha incominciato a distruggere la razza umana. Per fermarlo bisogna trovare ed usare le cinque macchine del tempo sul satellite, raggiungere cinque ere diverse e quindi recuperare i cinque pezzi di Amuleto posseduti da: *Mago Merlino, Leonardo da Vinci, Florence Nightingale, Giulio Cesare* e *Confucio*.

Il controllo avviene via mouse, clickando su apposite icone in basso sullo schermo, collocate sopra uno sfondo che varia di era in era. L'azione si svolge, con sprites animati, in una finestra relativamente piccola, dove il testo compare eventualmente nella porzione inferiore. I comandi si impartiscono, quindi, con molta semplicità ed immediatezza.

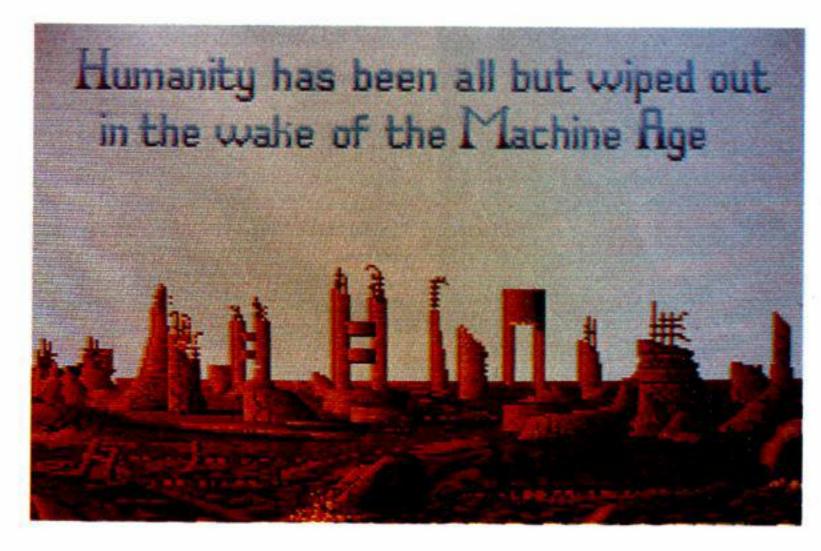
Vi sono circa cinquanta locazioni differenti nel gioco, ciascuna delle quali viene attraversata con uno scroll laterale continuo. Oltre a muoversi è possibile chiacchierare (che cosa racconteremo a Confucio?) tramite l'icona a *bocca*, afferrare, gettare, esaminare e tutto ciò che è normalmente concesso in un'avventura.

LA TECNICA

La grafica, in Time, svolge il proprio scopo egregiamente, anche se non è particolarmente eccezionale, specie se confrontata con quella di programmi come *Future Wars*.

Non mancano comunque alcuni tocchi pregevoli, rappresentati principalmente da alcune animazioni, come ad esempio la sequenza di atterraggio dello *Shuttle*. Gli effetti sonori sono limitati, in pratica quasi inesistenti.

IL VOTO

Un'avventura/arcade discreta, nè carne nè pesce, originale come trama ma realizzata con pochi preziosismi. 6 e mezzo.

Bisogna tornare indietro nel tempo per orientarsi tra alabarde, cavalieri e catapulte

Computer: Amiga inespanso
Gestione: Mouse
Tipo: Strategico
Softhouse: Arc

PRINCE

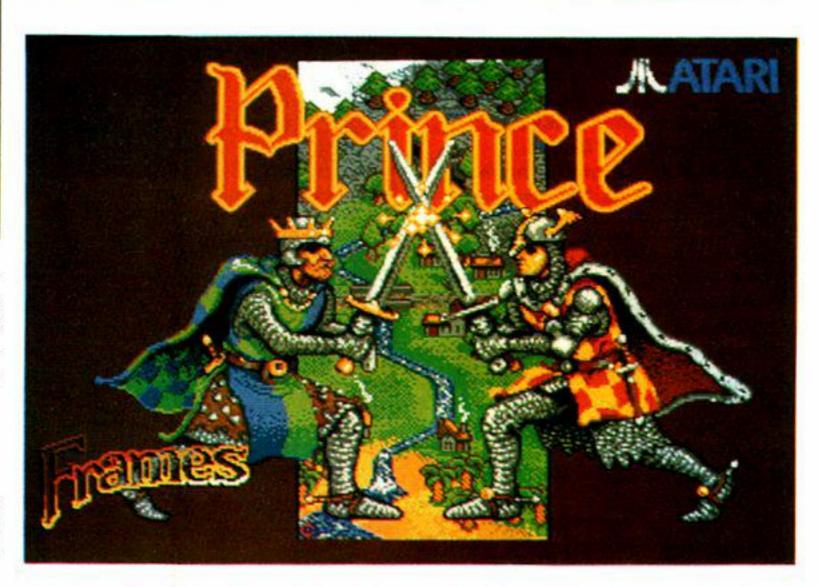
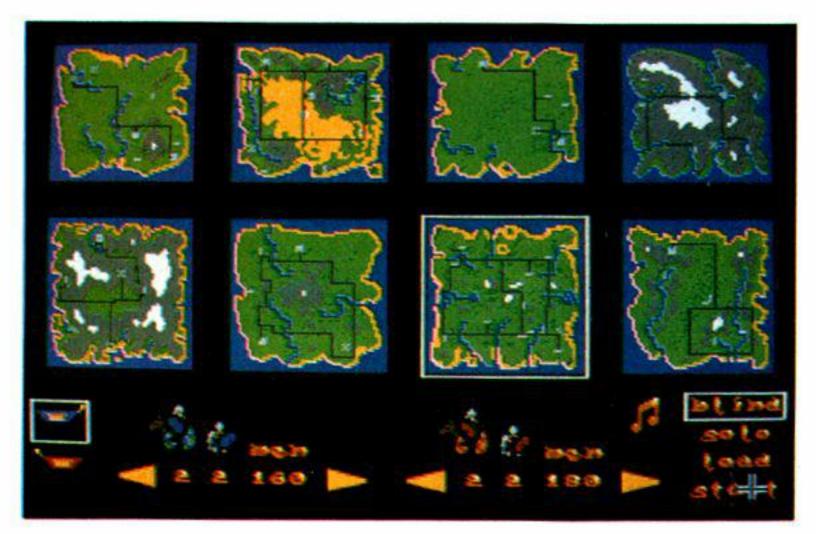
La simulazione di una guerra medievale, con distruzione dell'avversario a suon di colpi di alabarda, catapulte e mazze ferrate è l'obiettivo di questo programma strategico per uno o due giocatori.

IL GIOCO

Due prìncipi feudali di una piccola isola hanno deciso di farsi la guerra, quindi hanno ammassato un'armata composta da cavalieri e fanti.

Una volta fissata la dimensione di ciascuna armata (variabile tra i 45 ed i 514 uomini ciascuna) ed il tipo di isola (fra le otto previste) inizia il combattimento.

Sono previsti tre livelli di visibilità, basati sulla distanza tra gli eserciti, e quindi tre livelli di zoomata sul piano di gioco.

A seconda della selezione effettuata, gli ordini impartiti via mouse/gadget sono effettivamente trasmessi ad un reggimento o ad una sola compagnia. Per dare un ordine basta selezionare la compagnia od il reggimento, scegliere nella lista di comandi previsti *(turn, walk, trot, gallop, charge, reform, scout)* e spostare la sagoma della compagnia e del reggimento in un'altra locazione dello schermo. Quando due compagnie ingaggiano una lotta, questa prosegue sino a che uno dei prìncipi non muore; a questo punto, però, il gioco non finisce, in quanto non è più possibile produrre ordini, ma le altre compagnie procedono ad eseguire eventuali ordini precedenti e, se riescono ad uccidere il principe avversario, si perviene ad una patta.

LA TECNICA

La finestra di azione ricorda vagamente *Populous*. Al massimo livello di ingrandimento si vede una gran quantità di dettagli sul piano di gioco, con la realistica ondulatura del piano. Tutto è molto colorato e ben disegnato, sebbene diventi facilmente confuso se non si sceglie l'opzione *Remove Corpses*.

La musica di introduzione è ottima, mentre gli effetti sonori del gioco sono ben fatti, anche se ripetitivi.

IL VOTO

Un gioco divertente, tecnicamente ben fatto ma che richiede poca strategia per il genere a cui appartiene. 6+.

Un gioco semplice e divertente, destinato ad intrattenere gli utenti Amiga per lungo tempo

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade a piattaforma
Softhouse: Ormellysoft

SYSTEM 4

I giochi della generazione del leggendario *Q*Bert* non hanno evidentemente ancora fatto la loro epoca, grazie anche alla varietà ed alla qualità degli effetti resi possibili dai nuovi elaboratori.

IL GIOCO

Si controlla un simpatico robottino rimbalzante, *Mister Tengus*, che ricerca la sua fidanzatina al silicio ed acciaio (pure rimbalzante) su di una serie di piattaforme a gradini cubici.

Ovviamente sono previsti vari avversari, sempre rimbalzanti, che fanno perdere vite al robottino innamorato, e vari oggetti nutrienti che, se raccolti prima che scompaiano, consentono di sopravvivere (ma chi lo avrebbe detto che i robottini si cibassero di angurie, banane, caramelle...).

E' possibile giocare da soli, in coppia oppure in modo duale, dove un giocatore controlla l'avversario ed uno Mister Tengus.

Le piattaforme vengono completate saltando almeno una volta su tutti i gradini, che cambiano colorazione appropriatamente. Procedendo, le piattaforme diventano sempre più grosse e difficili da completare, sino alla più grande, che racchiude l'oggetto dei desideri.

Quando Mister Tangus si avvicina al bordo, si verifica un fluidissimo scorrimento laterale automatico.

E' da notare che il protagonista non può precipitare nel vuoto, nemmeno sbagliando a manovrare il joystick.

LA TECNICA

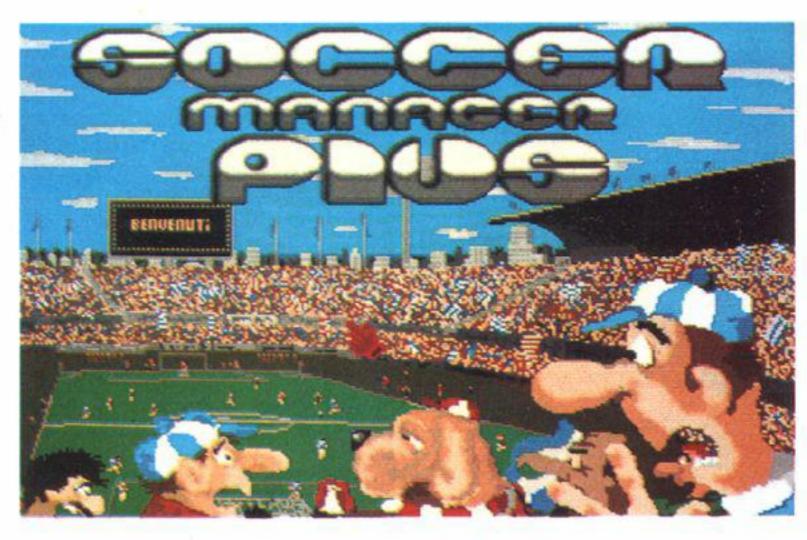
Nella sua semplicità, il gioco è certamente ben fatto. A molti può dare più soddisfazione giocare con un programma di questo tipo, scritto in meno di 100K di memoria, rispetto ad altisonanti giochi super sponsorizzati da parecchi mega distribuiti su vari dischi.

Lo scrolling non è nemmeno necessario, la grafica ben colorata e non stancante, la trama elementare e gli incentivi propri del tipo di gioco. La musica iniziale è semplice (come tutto il resto), gli effetti sonori ridotti, ma piacevoli.

IL VOTO

Forte nella sua semplicità, ben fatto. 7 più.

SOCCER MANAGER +

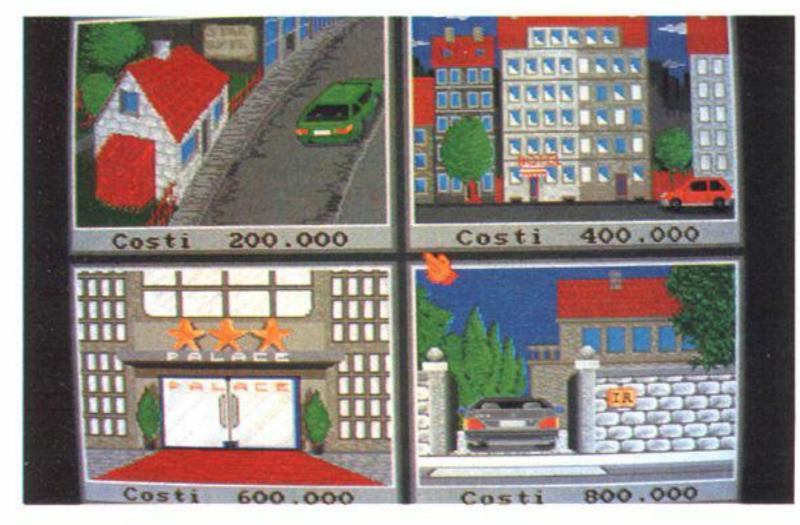
Si vestono i panni del vertice dirigenziale di una società calcistica: per un lauto stipendio(!) si deve affrontare un campionato scegliendo una squadra tra le trentasei dei campionati di serie A e B italiani.
Inizialmente si ha un contratto garantito per quattro stagioni, che può però essere annullato dalla società se la squadra retrocede o se si manda il bilancio in passivo.
Si parte comunque dalla serie B con venti giocatori, ciascuno dotato di vari parametri personali: età, ruolo, condizione fisica e valore sul calciomercato.
In omaggio al "calcio totale", tutti i giocatori possono comunque essere schierati in qualunque ruolo, basta solo che all'inizio di ogni partita vengano specificati, via mouse, dieci giocatori oltre al portiere e due riserve in panchina.
Si possono poi affrontare le varie partite eventualmente vedendo sullo schermo le semplici simulazioni animate delle *azioni gol* più salienti.
Tra un tempo e l'altro si possono modificare alcuni parametri di gioco della squadra e, ovviamente, sostituire i calciatori in campo con quelli in panchina. E' interessante notare che il programma assume che aumentando la "cattiveria" con uno slider si ottengono migliori risultati, ma anche un maggior numero di cartellini gialli e rossi. Le opzioni per gestire la stagione sono molteplici: si può fissare il prezzo del biglietto (se è alto guadagneremo di più ma il pubblico diminuirà, e viceversa), mandare la squadra in ritiro (costa, ma aumenta il rendimento ed è particolarmente utile se molti giocatori sono acciaccati), mettere all'asta i giocatori o trattare l'acquisto di quelli di altre squadre.

La Tecnica

Grafica esigua, considerato che le animazioni sono banalissime e lente risultano le scene di gioco a due dimensioni, mentre sono sufficientemente rifiniti i gadget di selezione. La veste grafica è piuttosto colorata, ma stanchevole. La simulazione dell'andamento del campionato, soprattutto dell'andamento delle partite, appare troppo semplicistica.

Il Voto

Un gioco per gli appassionati del genere strategico, ma solo per loro. 6 1/2.

Una nuova variante del ben noto videogioco classico Football Manager

Computer: Amiga inespanso
Gestione: Mouse
Tipo: Manageriale
Softhouse: Reflex

Christian Spanik

Amiga! Primi Passi

FTE Editions
Lire 40.000

La tendenza a non leggere il manuale di istruzioni fornito con la macchina è comune a tutti i possessori di Amiga. Lo sappiamo perchè notiamo che, spesso, le domande inoltrate alla Redazione dai nostri lettori meno esperti, troverebbero risposta chiara e semplice in esso.

Il manuale di istruzioni Commodore, tuttavia, non può certo essere sufficientemente ampio, dettagliato e "discorsivo" per soddisfare i principianti. *Amiga! Primi Passi*, che conta 225 pagine interamente dedicate all'insegnamento "passo per passo" dei rudimenti indispensabili all'utilizzo di Amiga può rivelarsi prezioso in casi come questi.

Nel *primo* capitolo vengono descritti sistematicamente tutti i collegamenti da effettuare (monitor o TV, eventuali floppy disk drive esterni, mouse, tastiera esterna, modulatore), con le differenziazioni del caso per possessori di Amiga 500, 1000 o 2000. In effetti tale lettura può essere molto istruttiva anche per chi "manovra" con Amiga da tempo. Quanti sanno che non si dovrebbe mai accendere o spegnere l'Amiga con i floppy inseriti nei drives, oppure come si regola il modulatore A590 per ottenere la migliore immagine o che cosa bisogna ritoccare se non si sente l'audio nel TV (a parte il volume)?

Nel *secondo* capitolo si descrivono, con parole molto semplici e comprensibili, che cosa rappresentano parole e concetti quali: Intuition, Workbench, Icone, Menu a tendina eccetera. Tutte queste cose vengono infatti spesso imparate per sperimentazione diretta, con le limitazioni e le possibili imprecisioni ed incompletezze che ne derivano.

Un perfetto novizio di Amiga, che legga scrupolosamente queste pagine, avrà certamente, alla fine, un concetto più chiaro e preciso di come funziona Amiga e quale sono le sue "filosofie" portanti di quanto può avere probabilmente un classico smanettone che cerca di imparare soltanto dagli errori o per sentito dire. Il capitolo prosegue spiegando dettagliatamente i contenuti dei vari cassetti del Workbench: Demos, System, Utilities.

Poi illustra l'uso del programma di configurazione Preferences, questo sconosciuto. In varie pagine vengono spiegate tutte le operazioni elementari necessarie per definire una sagoma di puntatore personalizzata, per scegliere e configurare una stampante specifica, per regolare i colori e la centratura dello schermo, per stabilire le dimensioni dei caratteri. Ovviamente viene anche spiegato passo per passo come si deve fare per registrare indissolubilmente queste informazioni sul disco, ovvero tramite il "click" del mouse sul gadget di Save, che registra le scelte fatte nel file System-configuration della directory Devs.

Il *terzo* capitolo è dedicato ai primi passi con il dischetto Extras, che contiene Amigabasic e vari programmi di supporto. I paragrafi spiegano come si lancia l'interprete, come si effettua il caricamento e il salvataggio di programmi dall'editor, come si usano i programmi dimostrativi e di supporto forniti sul disco dalla Commodore. Altri cinque paragrafi del capitolo sono dedicati a programmi didattici in Basic di grafica (illustra l'uso delle funzioni di base), animazione (Bob e Sprite), sintesi vocale (uso del **narrator.device**) ed un programma giocoso chiamato "Lotteria". Come sempre lo stile è molto piano e chiaro, mentre si spiegano come funzionano le varie istruzioni specifiche del Basic di Amiga, in modo comprensibili anche a chi ha scarsissima dimestichezza con il Basic. Ad esempio vengono spese tre pagine per spiegare, nel commento ad un listato, il funzionamento delle istruzioni If, Else, Endif con riferimento al programma riportato.

Il *quarto* capitolo è dedicato ad AmigaDos ed è molto succinto. Spiega comunque in modo chiaro che cosa è il **Cli** e ne mostra l'utilità ed i concetti di base con esempi d'uso elementari di istruzioni quali Cd, Dir, List, Rename e Delete. Tutto ciò consente al lettore inesperto di capire chiaramente che cosa è ed a che cosa serve la "Interfaccia a linea di comandi", senza ovviamente addentrarsi nello specifico.

Infine le appendici completano con informazioni interessanti il libro, dando cenni sulla manutenzione del computer e su che cosa si deve fare in casi di emergenza tecnica; quali sono i vari tipi di icone del Workbench e come si differenziano, un glossario di termini (che ormai usano tutti, ma spesso senza

averne un'idea precisa e corretta), come si usa l'editor di icone fornito dalla Commodore e vari dati tecnici.

Il volume sembra dunque consigliabile a chiunque abbia difficoltà ad impadronirsi dei rudimenti di base per usare Amiga. Il prezzo, relativamente contenuto (rispetto agli standard di mercato), la cura riposta nell'impaginazione contribuiscono a farne un buon acquisto.

Ruegheimer, Spanik

AmigaBASIC

FTE Editions
Lire 60.000 (con disco)

Nella stesura del libro AmigaBASIC i due autori, esperti programmatori tedeschi, hanno tenuto conto sia delle esigenze dei principianti che di quelle dei programmatori più esperti. Ne è nato un voluminoso "tomo" di ben **660** pagine, che tra l'altro non comprende molti dei listati di cui si parla, essendo stati inseriti esclusivamente sul dischetto venduto insieme al libro, per non portare a dimensioni da "Treccani" il libro stesso.

Tale impostazione dualistica si nota già sino dalle introduzioni, che sono due appunto, dedicate l'una ai neofiti e l'altra agli esperti. In una si spiegano i concetti di base minima per Amiga: come si usa il mouse, come si avvia l'interprete Basic, come funziona il Workbench eccetera, con uno stile di esposizione chiaro ed elementare.

L'introduzione per esperti riporta gli stessi concetti, ma con un linguaggio più sintetico e tecnico. In ogni caso il *primo* capitolo inizia spiegando dettagliatamente il funzionamento di tutti i menu del Basic, soffermandosi ad illustrare come si adoperano le varie funzioni richiamabili via mouse o combinazioni di tasti. Il tutto viene spiegato prima a livello teorico e poi, proseguendo nei paragrafi, con esempi pratici, scrivendo e correggendo passo a passo un semplice programma. Durante l'esposizione, a beneficio dei meno esperti, vengono illustrate anche specifiche del Basic come **Let, Dim, Val, Print** eccetera, rendendo la lettura più varia ed interessante.

Il *secondo* capitolo è dedicato al **colore**. In pratica si spiega dettagliatamente, con numerosi listati di esempio (sul disco) come si usano le funzioni grafiche e di controllo della tavolozza cromatica. Tra i programmi troviamo un generatore di grafici a torta ed uno che utilizza direttamente il Blitter, via AmigaBasic, per creare sagome di riempimento (fill pattern) di figure poligonali. Come sempre, il linguaggio adottato è aderente alla complessità degli argomenti: esempi e concetti semplici vengono spiegati con parole altrettanto semplici, mentre quando si parla di Blitter ed hardware, logicamente, il linguaggio diventa più tecnico, anche perchè si presuppone che il principiante si interessi prima alle cose semplici!

Il *terzo* capitolo parla della gestione in Basic di archivi su disco. Vengono illustrati i concetti di file, directory e device, ed illustrate le tecniche di gestione delle stesse in AmigaBasic. Come esempio viene mostrato l'ampliamento del programma di generazione di grafici a torta con l'inserimento della possibilità di leggere e scrivere archivi di dati su disco, oltre a listati di funzioni scritte in AmigaBasic per immagazzinare e gestire dati statistici.

Il *quarto* capitolo illustra dettagliatamente le tecniche di archiviazione e memorizzazione di grafici (istruzioni **Get** e **Put**) ed il formato universale di scambio di dati **IFF** (con routines di supporto in Basic), con applicazioni in programmi pratici e funzionanti.

Il *quinto* capitolo tratta in dettaglio l'accesso casuale a dati, ovvero illustra l'implementazione di un **Database** con le istruzioni di I/O del Basic. Ciò implica l'uso delle modalità di accesso relativo ad un file, ed i listati riportati illustrano in pratica come si può usare tale tecnica in AmigaBasic.

Il *sesto* capitolo sviscera i problemi di utilizzo pratico del **sintetizzatore vocale** interno ad Amiga, ovvero spiega come si usano i fonemi e le istruzioni **Say** e **Translate** del Basic. Dobbiamo dire che qui abbiamo trovato la trattazione più esauriente, da un punto di vista teorico / pratico, di quanto visto sinora in qualunque altra pubblicazione.

Il *settimo* capitolo è votato alla spiegazione della generazione di **musica** ed effetti sonori con AmigaBasic. Si impara cioè tutto sull'uso pratico di **Sound, Wait, Sound Resume**, concludendo alla grande in un ottimo programma chiamato **SinthAmiga**, che consente di fare della sintesi sonora da AmigaBasic.

Le numerose appendici (**120** pagine) sono spesso dei "supplementi" ad informazioni già date nel testo: I/O su schermo, animazione di oggetti grafici tramite le funzioni del Basic, comandi grafici in dettaglio, controllo del programma, funzioni di calcolo matematico, gestione evoluta della memoria di massa, descrizione sistematica dei comandi per la sintesi vocale e sonora.

Il volume rappresenta certamente un testo immancabile sulla scrivania di qualunque programmatore Basic di Amiga, sia esso un utente alle prime armi, che abbisogna di molti esempi pratici e della spiegazione di concetti di base della programmazione, sia esso un evoluto programmatore, che deve comunque "ispirarsi" a routines e funzioni di gestione avanzata. Il prezzo, considerato la quantità enorme di informazioni e la presenza del dischetto, è più che onesto.

R. Burger

Virus

FTE Editions
Lire 40.000

I virus sono, come molti sanno, dei programmi che si installano e si riproducono ad insaputa degli utenti di computer, provocando effetti collaterali nocivi o, comunque, non voluti. Esistono da quando esistono gli elaboratori elettronici e ven-

gono scritti, tipicamente, da persone molto esperte in linguaggio assembler per vari scopi: scherzosi ed innocui, deleteri o come eliminatori di altri virus. Alcuni sono *specializzati* per potersi riprodurre via **BBS**, o per inserirsi sempre nei bootblock dei dischetti in computer come Amiga, oppure per installarsi nel DOS su disco in computer come gli MS/DOS, oppure ancora per installarsi vita natural durante sui dischi rigidi dei sistemi.

Non vengono colpite da epidemie virulente soltanto macchine particolarmente esposte come Amiga, ma anche reti di terminali di università, di banche o della NATO, come recenti episodi di cronaca ci hanno fatto sapere, che dovrebbero invece essere super-protetti.

Abbiamo detto che Amiga è particolarmente esposto ai virus per vari motivi, spiegati nel libro di Burger: sistema operativo multitasking, capacità di autoboot da dischetto, ampio mercato di programmi piratati (spesso portatori di virus) e cospicua presenza di BBS aperte a tutti e molto "indifese", essendo gestite da personale (e software, quindi) inefficiente sotto il punto di vista della autodifesa.

Il libro di Burger, comunque, illustra dettagliatamente il modo di funzionamento dei virus in vari sistemi operativi, compresi UNIX e MS/DOS, riportando addirittura dei **listati esemplificativi di come sono scritti**. Ovviamente vengono presentati solo virus ben noti e considerati "innocui" in quanto esistono schemi e programmi di difesa sufficienti per annientarli!

In effetti Burger illustra il problema non solo da un punto di vista prettamente tecnico, illustrando concetti come "intercettazione delle chiamate alle routines di scrittura su disco", "manipolazione di software", "bombe a tempo", "link-virus", "virus dormiente", "bootblock virus" ed altri, ma anche dal punto di vista dell'utente medio, interessato a conoscere i retroscena storici e culturali del problema, senza esser costretto a studiare listati assembly.

Il libro, dopo le consuete introduzioni all'edizione tedesca ed all'edizione italiana, parte con un *primo* capitolo dedicato alla spiegazione di cosa è un virus e di come funziona, dando anche un glossario preciso di termini relativamente "nuovi", utili come riferimento quando si vuole parlare dell'argomento con gente competente senza timore di essere fraintesi.

Il *secondo* capitolo è certamente il più interessante per i non-tecnici, dando ampi cenni storici e di cronaca sul fenomeno. Vengono riportati qui gli studi di Fred Cohen, un famoso informatico che ha eseguito parecchi studi su questo tipo di programmi.

Il *terzo* capitolo illustra i pericoli derivanti dai virus, spiegando con esempi e cenni di cronaca come si può propagare e come vengono usati (o possono potenzialmente essere usati) come strumenti di guerra software; si pensi che l'intera rete difensiva del Patto di Varsavia e della NATO sono gestite tramite reti di computer, che possono essere infettate...

Il *quarto* capitolo è ancora di cronaca e tratta in particolare del congresso **Chaos** tenuto ad Amburgo dove oltre trecento informatici discussero delle ricerche sui virus e sui metodi di neutralizzazione. Sulla stessa trama è anche il *quinto* capitolo, dove si spiega, con cenni di cronaca, come gli utenti comuni siano indifesi rispetto alle epidemie virulente, e quindi terreni di cultura e di propagazione inconsapevole.

Il *sesto* capitolo è più strettamente tecnico perchè riporta listati di virus di vari tipi: generatori di errori software, aggressori dell'hardware, riduttori delle prestazioni ed altro ancora. Viene anche spiegato come i virus siano stati usati per ricatti ed estorsioni a danni di aziende utilizzanti reti di terminali.

Il *settimo* capitolo parla dei metodi e delle filosofie di protezione contro i virus. Se è molto difficile pensare di proteggere globalmente un sistema o del software contro un virus, certo si possono ridurre i rischi di contaminazione, inserire dei meccanismi di "sterilizzazione" o riconoscimento di infezioni o dei terminali a rischio e così via.

Si danno anche cenni di come deve essere strutturato un sistema operativo ideale antivirus e di come ci si può assicuare con agenzie contro pericoli di questo tipo.

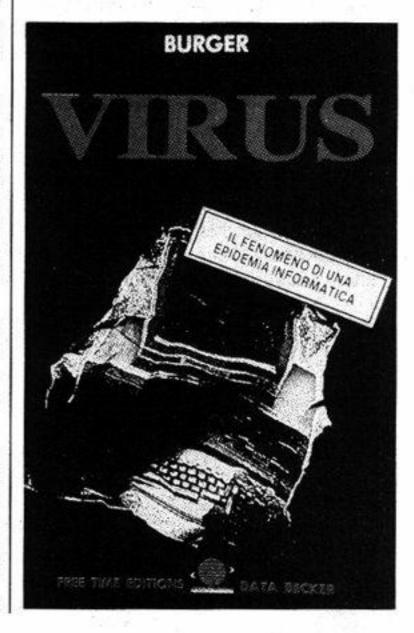
L'*ottavo* capitolo riporta listati di virus di vari tipo: residenti, di sovrascrittura, di non sovrascrittura, a richiamo, mentre il *nono* capitolo riporta listati di virus in linguaggio Assembler, Pascal, Basic e Batch. Il *decimo* capitolo illustra invece in dettaglio come vengono tipicamente infettati i sistemi operativi **CP/M**, **MS/DOS** e le **reti**.

L'*undicesimo* capitolo parla dei virus che si propagano per rete telefonica (**BBS**) ed erudisce i programmatori sul come si può ovviare a questi rischi.

Il *dodicesimo* capitolo mostra di conseguenza alcuni schemi di protezione pratica di dati e servizi multiterminale, mentre il *tredicesimo* capitolo spiega quali sono le azioni distruttive tipiche portate dai virus, consentendo di riconoscerli.

Il *quattordicesimo* capitolo illustra in circa *cinquanta* pagine quali sono le tecniche di protezione pratica contro i virus: sistemi operativi antivirus, protezione mediante automutilazione, programmi cacciatori di virus, programmi antivirus, ricercatori di alterazioni, eccetera.

Il *quindicesimo* capitolo parla delle possibili evoluzioni del problema, con un occhio al possibile software del futuro e, quindi, dei virus del futuro.

Concludendo possiamo affermare con certezza che il libro, di **380** pagine, fornisce una grande quantità di informazioni originali e complete. Inoltre, particolare degno di nota, spesso tali informazioni non sono strettamente tecniche, quindi facilmente leggibili e comprensibili anche a chi non è un programmatore esperto, dato anche lo stile di scrittura molto piano e comprensibilie.

Prentice Hall

M68000 User's Manual (6th edit)

Motorola Inc.
Lire 70.000 (circa)

Questo libro è **IL** testo di riferimento fondamentale per chiunque voglia programmare in linguaggio *Assembler* i microprocessori della famiglia Motorola **MC 680x0**. La sesta edizione tratta esplicitamente i microprocessori 68000 (quello di Amiga), 68008 (quello del Sinclair QL), MC68010 e MC68HC000. Si tratta di un testo strettamente tecnico, ma contenente informazioni indispensabili ed altrimenti introvabili per chi desidera programmare o lavorare seriamente con questi processori.

Il libro è di circa **300** pagine, scritte in modo piuttosto fitto con molti grafici e diagrammi di temporizzazione, e suddiviso in **sedici** capitoli e **cinque** appendici.

Dopo il capitolo *introduttivo*, si passa alla descrizione delle strutture hardware interne del microprocessore. Qui si spiegano concetti cardine della microinformatica applicata, come i concetti di programmazione modulare strutturata, consistenza di strutture, possibilità di verifica del software, macchina a memoria virtuale e macchina virtuale. Tutto ciò sta alla base della filosofia di sviluppo del processore della Motorola.

Il *terzo* capitolo spiega dettagliatamente l'organizzazione dei dati del microprocessore e le sue capacità di indirizzamento. Sono previsti ben 36 sottoparagrafi, ciascuno illustrante un solo argomento, per consentire un rapido riferimento al programmatore Assembler durante lo sviluppo del proprio software: dimensioni degli operandi, indirizzamenti a registri, indirizzamento effettivo, indirizzamento assoluto e relativo, manipolazione dello stack e tutto ciò che ne consegue. Il linguaggio è tecnico, ma mai eccessivamente succinto e sintetico, certamente chiaro per chiunque abbia le basi necessarie per studiare questi argomenti.

Il *quarto* capitolo è un sommario degli insiemi di operazioni possibili: spostamento di dati, aritmetiche, shift e rotazione, logiche, manipolazione di bit, operazioni BCD, controllo del programma, controllo del sistema, controllo di mutliprocesso. Le istruzioni sono tutte riportate in chiare tabelle di consultazione con dimensioni degli operandi, note d'uso, tecniche di indirizzamento concesse e tutto quanto serve a chi ha bisogno per una consultazione.

Il *quinto* capitolo è dedicato ai tecnici elettronici ("hardwaristi"), dato che spiega il funzionamento del microprocessore 68000 come circuito elettronico, piedino per piedino e segnale per segnale. Le pagine sono costellate di diagrammi di temporizzazioni che illustrano come variano i segnali sui piedini del processore allo scorrere dei segnali del clock, durante le esecuzioni delle varie istruzioni. I diagrammi sono essenziali per chiunque deve sviluppare anche una semplice interfaccia (od un computer!) usando i processori della Motorola.

Purtroppo si arriva sino alle descrizioni del 68010, mentre gli utenti di sistemi con 68020 o 68030 dovranno consultare altri testi.

Il *sesto* capitolo è dedicato esclusivamente al trattamento delle *eccezioni*. Qui vengono sviscerati argomenti apparentemente difficili ed impalpabili come: modo supervisore/utente, cambiamenti del modo di privilegio, reset, interruzioni, interruzioni spurie, sequenze di trattamento delle eccezioni, trappole, istruzioni illegali e non implementate, violazione di privilegio, tracciatura, errori di bus e di indirizzamento. Anche questo capitolo contiene una mole di informazioni fondamentali per chiunque voglia programmare un computer con 68000, come Amiga. Per intenderci, si può anche imparare a memoria lo *Hardware Reference Manual* di Amiga, ma se non si hanno chiari i concetti qui esposti non si riuscirà a fare mai nulla di buono.

Del resto tutti i libri che spiegano, o cercano di spiegare, i concetti posti alla base del funzionamento dell'elaboratore, fanno riferimento ai capitoli di questo libro, scritti da mamma Motorola in persona.

Il *settimo* capitolo è molto breve ma intenso, trattando delle modalità di trasferimento elettronico dei dati del processore e delle operazioni necessarie per interfacciare le interruzioni.

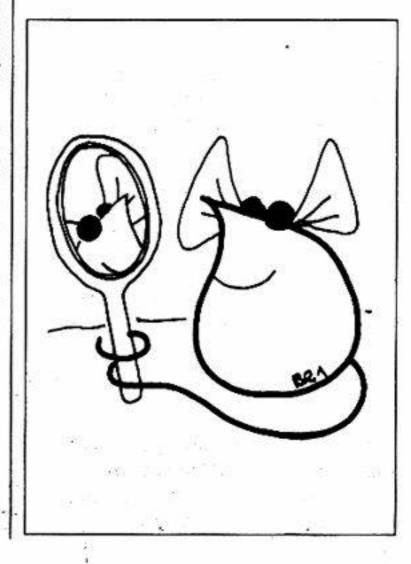
L'*ottavo* capitolo è ancora uno di quelli vitali per i programmatori Assembler, dato che riporta le temporizzazioni di tutte le istruzioni del processore 60000. Le tabelle di riferimento sono effettivamente indispensabili quando si vuole ottimizzare un programma o scrivere routines sensibili alla temporizzazione.

La suddivisione è in classe di operazioni svolte e le informazioni sono date, come sempre, in modo elegante e preciso. Il *nono* ed il *decimo* capitolo sono repliche dell'ottavo, ma riferite al processore 68008 e 68010.

L'*undicesimo* capitolo riporta le specifiche tecniche del processore 68000, con un occhio particolare all'aspetto "elettrico", vitale per i progettisti elettronici: prestazioni, caratteristiche termiche, considerazioni sull'alimentazione elettrica, caratteristiche elettriche, temporizzazioni, cicli di lettura e scrittura in memoria e gestione del bus. Il *dodicesimo, tredicesimo* e *quattordicesimo* capitolo sono repliche dell'undicesimo, dedicati però rispettivamente ai processori 68HC000, 68008 e 68010.

Il *quindicesimo* ed il *sedicesimo* capitolo riportano informazioni più strettamente commerciali sul processore: package, modalità di ordinazione alle case produttrici, disponibilità, eccetera.

Le *appendici* spiegano: il calcolo, i codici di condizione dopo ciascuna istruzione, il funzionamento di tutte le istruzioni del set 68000, i formati delle istruzioni e note sul modo looping del 68010. Chiude il libro un completo *indice*.

Il volume è ovviamente consigliatissimo a **chi sa l'Inglese** e vuole avere il testo originale a cui fanno riferimento tutti gli altri libri che parlano di programmazione Assembler del 68000.

AMIGAZZETTA 9, DA OGGI ANCHE IN BBS

Più comodo, più facile; ecco ciò che promette la nuova raccolta software per i fedelissimi di CCC

by Domenico Pavone

E' ormai assodato che il Public Domain, ormai anche per Amiga di ragguardevole quantità (oltre che qualità), ha assunto in pieno l'impegnativo compito di supplire a tutte quelle piccole imperfezioni, deficienze o scomodità non (ancora) risolte dai progettisti al soldo della Commodore.

Per non parlare della possibilità di disporre, senza esborsi talvolta troppo onerosi, di programmi che spesso non hanno nulla da invidiare ai prodotti di solide software house.

Un esempio?

Ma i files inclusi in questo **Amigazzetta**, naturalmente!

Stavolta si è esclusa un po' la fascia dei "programmers", per prediligere una serie di utility davvero molto comode, di quelle che, una volta provate, non si può più farne a meno (occhio all'overdose di Guru, però).

Cominciamo subito con la prima, che affronta brillantemente proprio il più odiato, e purtroppo frequente, nemico degli amighi.

NOGURU

Al momento, è senz'altro il più valido strumento per debellare i danni prodotti dalle famigerate Guru Meditation, ivi compresi gli stressanti stati d'ansia di indifesi users.

Tutto ciò che occorre fare è aprire il cassetto **Programmi/Noguru** di Amigazzetta, e biclickare su **NoGuru**.

Apparentemente non accadrà nulla, ma il programma sarà già installato in memoria, pronto ad intervenire.

Per verificarne il funzionamento, si può provocare intenzionalmente un blocco del sistema attivando una delle due icone incluse nella directory (Guru1 e Guru2).

Al posto della solita banda nera, apparirà una schermata dello stesso colore, ma con due possibilità di scelta: se ci si trovava in ambiente workbench (come nel nostro caso), occorre premere il tasto destro del mouse, e... magia: rieccoci allo schermo normale, senza il deprecato reset del sistema!

Analogo risultato si otterrebe con il tasto sinistro, che però è consigliabile se, al momento del guru, era attiva una finestra shell / cli.

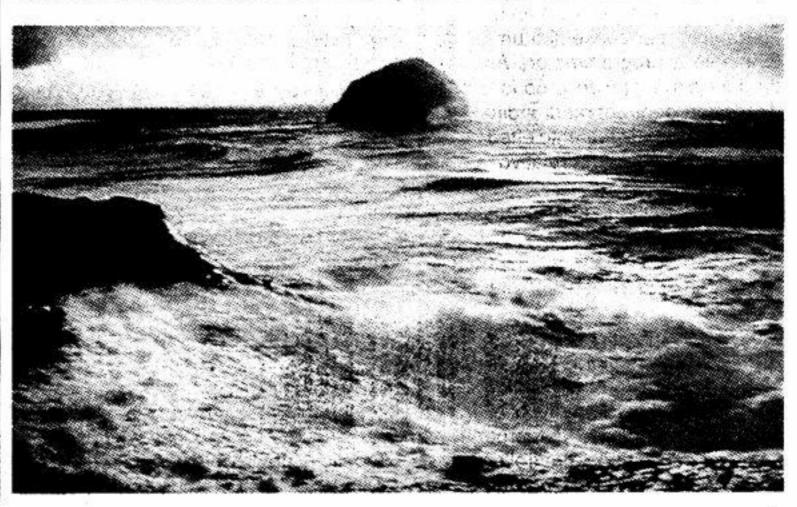
In quest'ultima eventualità, il programma tenta di aprirne una nuova, a patto che la directory **C** contenga il comando **NewCli**.

In entrambi i casi, comunque, è meglio non fidarsi troppo dei capricci di Amiga e salvare subito il salvabile qualora si stessero gestendo dati importanti: ulteriori blocchi potrebbero diventare irreversibili anche per NoGuru, che comunque rappresenta un notevole passo avanti rispetto al passato, magari in attesa di miracoli in versione 1.4...

POPUPMENU

Ovvero, come liberarsi dalla schiavitù della barra di comando.

Quella, per intenderci, sulla quale occorre portare il puntatore del mouse prima di premerne il pulsante destro, per accedere ai vari menu di un programma o dello stesso Workbench.

Anche in questo caso, basta attivare come di consueto la relativa icona, e verificare immediatamente il funzionamento del programma premendo il pulsante destro del mouse in qualunque

parte dello schermo: i menu appariranno subito a ridosso del puntatore, orientati verticalmente, consentendo una scelta immediata e "sul posto", senza sforzare troppo i delicati muscoli delle nostre preziose braccia.

Per disattivare PopUpMenu, è sufficiente mandare in esecuzione una seconda volta il programma.

NOCLICK

Altro minifile cui affidare la nostra riconoscenza, stavolta a tutto beneficio di una differente sezione anatomica: l'orecchio. Che, pur non essendo direttamente impegnato nell'uso del computer (a meno di strane ed improbabili contorsioni sulla tastiera), è spesso frastornato dal continuo "click" emesso dai drive senza floppy.

Certo non è un rombo di tuono, ma la silenziosa concentrazione di un computeromane, magari già messa a dura prova dal ronzio della ventola di un A2000, è molto sensibile agli sbalzi acustici.

Ebbene, con Noclick il problema è risolto: solita doppia pressione di **LMB** (per gli amici *Left Mouse Button*, ovvero il pulsante sinistro del mouse), un paio di secondi di attesa, ed il gioco è fatto.

I più esigenti, ed è il miglior utilizzo possibile, potranno inserire Noclick come uno dei comandi della Startup-sequence.

BROWSER

Questo programma, in un certo senso, potrebbe annoverarsi nella categoria delle cosiddette **Dir Utility** (Climate, Sid, DirMaster, ecc.), ma in effetti è qualcosa di più. Il suo utilizzo, infatti, rappresenta un valido aiuto tanto in ambiente Workbench che Shell.

Nel primo caso, consente infatti di manipolare qualunque file, praticamente con le stesse note modalità, anche quando questo non è fornito di icona.

Viceversa, trovandosi a smanettare da una finestra Dos, si dispone delle stesse comodità offerte dal Workbench, evitando molte noiose digitazioni.

Per verificarlo, si agisca due volte col mouse sull'icona di Browser: apparirà una finestrella ridimensionabile con elencate tutte le periferiche attive, tanto fisiche che virtuali.

Ora, si provi a clickare due volte (p. es.) su **C:**. Si aprirà una nuova finestra, con la descrizione di tutti i files della directory C.

Non pensate che sia finita qui, il bello deve ancora venire. Si provi a ripetere l'operazione (nella prima finestra di Browser) biclickando su **Ram Disk:**.

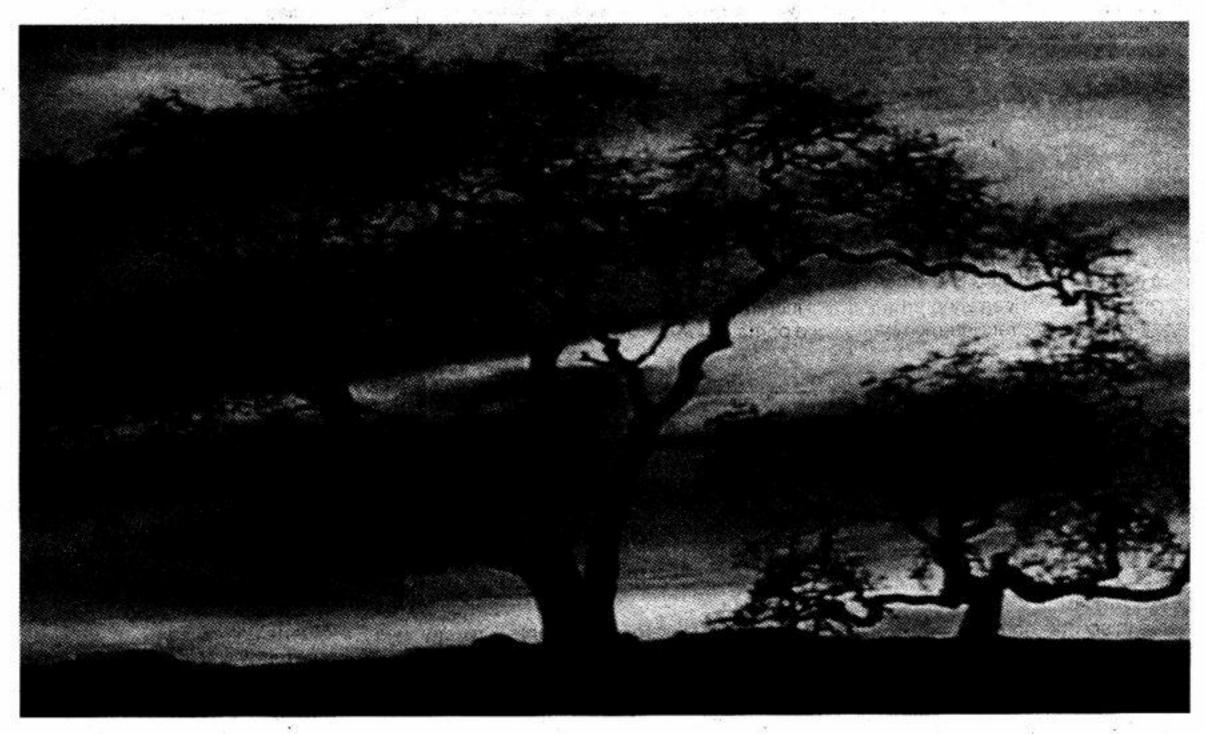
Ora, nella finestra di C:, si selezioni un file a caso con il mouse, premendo il pulsante sinistro solo una volta e, mantenendolo abbassato, si porti il puntatore all'interno della finestra RAM:.

Il **file verrà copiato** in Ram Disk, proprio come si fa in Workbench trasportandone la relativa icona.

Con le stesse modalità si possono selezionare più files adoperando lo Shift assieme al pulsante sinistro del mouse, e, come ovvio, le directory interessate agli spostamenti possono essere le più svariate, visibili o meno che siano sullo schermo Workbench.

Molte altre opzioni (tutte piuttosto intuitive) sono poi presenti nella **Menu Bar**, selezionabili (tramite il consueto tasto destro del mouse) quando almeno una delle finestre di Browser è attiva.

Oltre ai ben noti **Rename, Delete**, eccetera, molto utilili risultano inoltre le scelte **Wb Tool** e **Cli Tool**, che però richiedono una certa attenzione.

Per capirne il funzionamento, ricorriamo ancora ad un esempio.

TELEFONO AMIGO

Nelle pagine della rivista, come certo vi sarete accorti, è cominciato ad apparire con sempre maggiore frequenza il logo **BBSYSTEMS**, la nostra (e vostra) banca dati, raggiungibile **24 ore su 24** con l'ausilio di un modem e una semplice telefonata.

Non appena si uscirà dalla fase sperimentale, e al momento in cui leggerete queste righe la cosa dovrebbe già essere avvenuta, tutto il materiale finora inserito nei 9numeri di Amigazzetta sarà disponibile al cosiddetto Download (prelevamento via modem).

Il che significa che, senza stare ad ammattire con i disservizi delle nostre amate(?) Poste, comodamente adagiati (beh, non esageriamo) davanti alla nostra tastiera, si potrà disporre "in tempo reale" tanto dello smisurato materiale di Pubblico Dominio che circola per Amiga, quanto (se non soprattutto) di tutti **i listati pubblicati nella rivista**.

Quindi niente più lunghe digitazioni dagli esiti spesso frustranti, per non dire della possibilità di inoltrare in modo rapido e sicuro qualunque vostro quesito, consiglio, o contributo.

In quest'ottica, il supporto magnetico di Amigazzetta diverrà sempre più obsoleto, per due semplici motivi.

Anzitutto, per una considerazione di carattere economico.

Se, infatti, è vero che la Sip non fornisce gratuitamente i suoi (non brillanti) servigi, basti pensare che tutti i files di un Amigazzetta, da noi debitamente compattati, possono essere prelevati gratuitamente con un collegamento telefonico di 10-15 minuti a 2400 baud.

E si è parlato di **tutti** i files di un disco; selezionando solo quelli che interessano, i tempi si accorceranno di conseguenza.

Ora, se si paragona il costo di un tale collegamento con la cifra necessaria per l'acquisto del floppy e delle relative spese di spedizione, il vantaggio risulterà ovvio anche per chi risiede in latitudini opposte rispetto alla sede della BBS.

Inoltre, per motivi di spazio, il dischetto di solito non contiene le documentazioni originali allegate ai files, che saranno invece regolarmente associate ai programmi compattati nell'ambito di BBSYSTEMS.

Di solito si tratta di testi in inglese, ma comunque più completi di quanto possa essere succintamente descritto nelle recensioni di queste pagine, per non parlare della frequente presenza dei listati sorgente in **Basic**, **C** oppure **Assembly**.

Tra l'altro, i lettori che eventualmente si dedicassero ad una loro traduzione, o comunque ad altro tipo di contributo pratico al Bullettin Board (la banca dati, per intenderci), verranno premiati con incrementi di privilegio nell'accesso, o altre forme che renderemo note quanto prima.

Nella finestra **C** prima aperta, si selezioni con un click (uno solo!) il file More, magari agendo sulla barra laterale per rintracciarlo, o aumentando le dimensioni della finestra. Poi, dalla Menu Bar, si scelga Wb Tool. Da questo momento, sempre nella Title Bar, sarà disponibile un nuovo Menu, di nome Tools, contenente la scelta "more".

Per utilizzarla, basterà selezionare il file che si intende leggere, e scegliere questa opzione.

Si provi, ad esempio, a biclickare (finestra Browser) su Amigazzetta9 (in arancione), selezionare il file Clickami! (un solo click), e scegliere More dal menu principale.

E' proprio tutto un altro andazzo.

Unica precauzione, non assegnare come WB Tool un file che funzioni solo in Cli (o Shell): per questi serve l'opzione Cli Tool, che provvede automaticamente ad aprire una finestra Dos, ed eseguire il comando selezionato.

Così come in Workbench, qualunque file può inoltre essere lanciato biclickando sul suo nome.

SMARTICON

Si tratta di una simulazione del classico **Iconize**, spesso implementato in molti programmi di un certo livello.

Una volta mandato in esecuzione (anche da Workbench), il suo effetto è immediatamente visibile: tutte le finestre, presenti e... future, vengono automaticamente dotate di un nuovo gadget posto tra i due di profondità.

Clickandovi sopra, la finestra apparentemente scompare, ma è solo posta in stato di "latenza", pronta ad essere riaperta in qualunque momento accedendo come di norma alla Ram Disk (apertura della sua icona) e biclickando col mouse sull'icona che Smarticon avrà creato al suo interno. Come ovvio, l'unica finestra da non iconizzare in nessun caso è quella della Ram Disk.

FMC

Con questa utility, è possibile controllare lo stato della memoria di Amiga, con in più la facoltà di "switchare" tra Chip e Fast Ram (per chi ce l'ha) in modo molto più immediato e trasparente che non con il tradizionale NoFastMem. FMC apre una piccola finestra con ben in evidenza un interruttore, la cui posizione viene modificata da un "colpo" di mouse.

L'ammontare della memoria disponibile viene sempre chiaramente mostrata al suo interno, a meno che non si iconizzi la finestra premendo il pulsante destro del mouse quando questa è attiva.

Una ulteriore pressione di **RMB** (Right Mouse Button) rivisualizza FMC, consentendo eventuali nuove modifiche alla priorità di accesso in Ram.

SOUND TRACK UTILITY (STU)

Consente di prelevare, modificare, ed eventualmente salvare, brani musicali in formato **Sound Tracker** presenti in memoria, vale a dire dopo una loro precedente esecuzione.

Esistono in circolazione anche altri programmi del genere, ma STU si distingue per una interfaccia utente estremamente "friendly", tutta basata sulle facilitazioni offerte da Intuition.

In altre parole, si può far tutto con qualche clickata del mouse, senza complicazioni digiterecce.

LA DIRECTORY CCC

Come ormai consuetudine, Amigazzetta contiene una directory riservata a quanto pubblicato su Commodore Computer Club.

Il numero della rivista cui fanno riferimento i programmi, può essere desunto biclickando sull'icona **Indice_CCC**, che consente di consultare un sommario generale di tutti gli articoli riguardanti Amiga.

Più direttamente, la stessa informazione viene fornita dalla finestra Info del programma (o della directory), visionabile attivandone l'icona con un singolo click e selezionando Info dal menu Workbench (da Shell, la nota sarà presente come commento dopo un **List CCC**). Si tenga presente che nel disco, per ovvi motivi di spazio, non è memorizzato l'interprete AmigaBasic.

Se necessario (sarete avvisati da un Requester), può quindi rendersi obbligatorio inserire **Extras 1.3** nel drive (se lo si possiede, in quello esterno).

Si presti particolare attenzione, se non si è molto esperti, nelle eventuali risposte agli input richiesti dai programmi, quando questi richiedono l'immissione del nome di files. Volendo, per esempio, lanciare il demo contenuto in ConverBrush, è opportuno prima impartire nella finestra di output del Basic...

Chdir "Ccc/ConvertBrush"

...pena un inevitabile File Not Found.

Per i programmi che richiedono l'ausilio dei files **.Bmap**, questi ultimi sono già inseriti nella directory **Libs** di Amigazzetta 9.

I files eseguibili da Shell (o Cli), sono invece memorizzati nella directory **C**; la loro icona, se attivata, produrrà la semplice apertura di una finestra Dos, dalla quale lanciare il programma digitandone il nome (eventualmente preceduto da Run).

Tutti gli eventuali sorgenti, sono visionabili biclickandone l'icona; per le istruzioni del programma di lettura (peraltro intuibili), si prema il **tasto Help** in qualunque momento della visualizzazione, tenendo presente che LMB indica il pulsante sinistro del mouse, e RMB quello destro.

Per apprezzarne le possibilità, si provi (memoria permettendo, con certi brani 1 Megabyte di Ram è pressochè indispensabile) ad ascoltare uno dei demo presenti nella directory **Music Grab** (in **CCC**), poi, dopo esserne usciti (con il mouse o il joy, a seconda del Demo), si mandi in esecuzione **STU**.

Per una descrizione più approfondita del programma, si veda l'articolo della rubrica Assembly dedicato alla manipolazione della memoria in rapporto alla musica.

LOLLYBOBS

A che cosa può servire un demo? A niente, dirà qualcuno.

Eppure, se il risultato lo merita, talvolta è bello lasciarsi ipnotizzare dalle stupende capacità grafico - musicali di Amiga.

Questo programma, creato da un gruppo di softwaristi "d'assalto" made in Germany, ne è un esempio.

Lo si mandi in esecuzione, poi si abbassi il pulsante sinistro del mouse dopo l'apparire della prima schermata, e lo stesso si faccia dopo la comparsa di una banda orizzontale multicolore.

Da questo momento in poi, si agisca solo sul pulsante destro per modificare il tipo di animazione visibile (col sinistro si esce dal demo).

Prima di abbandonare lo schermo, si provi però un'altra via: il reset (**Ctrl + A + A**).

Altri **180 Bob** creeranno delle armonie grafiche forse più belle delle precedenti, anche qui modificabili col pulsante destro del mouse (almeno quattro possibilità di scelta).

Lo sforzo di programmazione degli amighi tedeschi è certo rivolto a meri fini pubblicitari, ma, visti i risultati, decisamente meritati.

AZCOMM 2.5

Per concludere, considerati gli sviluppi telematici (vedi riquadro) ormai consolidati, non poteva mancare un valido programma di comunicazione, basato sul più affidabile e veloce protocollo di trasmissione dati: lo **ZModem**.

Se avete appena acquistato un modem (ma anche se già l'avete), AZComm è il modo migliore per provare a collegarvi con la nostra **BBSystems**.

Il programma, semplice ed essenziale, "supporta" automaticamente la **grafica Ansi**, consente qualunque velocità di trasmissione, lo **Split Screen** (divisione dello schermo per digitare a vista ciò che si vuole) e, soprattutto, è molto semplice da usarsi. Dispone, inoltre, di un "elenco telefonico" che appare all'estremità destra della barra principale. Per aggiornarlo, è necessario inserire i nuovi numeri nel file **Comm.phone**, memorizzato nella stessa directory Programmi, ma non visibile da Workbench.

Occorre quindi aprire una finestra Shell, impartire CD Programmi, ed infine ED Comm.Phone:

A questo punto, è sufficiente accodare alle due "Entry" già presenti (quelle di BBSystems) nome, telefono, velocità di trasmissione (ed eventuali commenti) di altre banche dati, rispettando la stessa spaziatura adoperata per il numero della nostra.

A proposito di quest'ultima, si tenga presente che è registrata due volte, per contemplare la possibilità che si chiami dallo stesso distretto telefonico oppure in teleselezione.

In **BBSYSTEMS.02**, il numero telefonico (**5249211**) è fatto precedere dal prefisso teleselettivo 02, mentre la voce BBSYSTEMS del "Phone Book" è sprovvista del prefisso.

La velocità è settata su 1200 baud, ma, se il modem lo consente, può essere incrementata a 2400.

Può anche essere ridotta a 300, ma l'ipotesi è pressochè da scartare nel caso dei files per Amiga, spesso di dimensioni proibitive (in termini di tempo) per i 300 baud. Per memorizzare il file, come si dovrebbe ben sapere, basta uscire da ED premendo **Escape**, poi il tasto **X** ed infine Return.

Si badi che i primi caratteri del file rappresentano alcuni codici Hayes, eventualmente modificabili da chi già ha una certa dimestichezza con i modem.

Gli altri si limitino a non alterarli, almeno fino a che non saranno in grado di farlo.

Appuntamento a presto, dunque, ed un arrisentirci via modem.

CAMPUS
AMIGA

AMIGAFACILE, OVVERO I COMANDI AMIGADOS UNO PER UNO

A partire da questo numero *Commodore Computer Club* comincia a descrivere in modo approfondito i comandi Amigados, talvolta nebulosi e criptici all'inverosimile.
Tale iniziativa è volta a coinvolgere tutti gli utenti, normali o esperti che siano, in modo da chiarire le lacune lasciate dai manuali (in inglese) troppo spesso approssimativi.

ASSIGN

COPY

DATE

INSTALL

DIR

PATH

SEARCH

SORT

78 - UNA CORSA A BORDO DEL JOYSTICK

Grazie ad un listato piuttosto breve in Amigabasic possiamo costruire un simpatico giochino ideale per gli "smanettoni" del joystick.
In perfetto stile olimpionico-elettronico potrete costruire il videogame adatto sia per imparare Amigabasic sia, perchè no, per puro divertimento.

Nessun albero viene abbattuto per gli inserti di *Commodore Computer Club*, stampati su carta riciclata al 100%

ASSIGN

Assign. Impartito da solo, o seguito da **List**, produce un elenco degli assegnamenti al momento attivi.

Percorso. Specifica il nome della directory o del file (eventualmente fornito di path completo) da assegnare al device logico. Non necessita dei due punti finali, a meno che non si riferisca al nome di una periferica fisica (**Df1:**, **miodisco:**, eccetera)

List. Opzionale, pressochè superfluo. Aggiunto subito dopo il comando, svolge la stessa funzione di **Assign** da solo. Posto dopo i parametri **Device:** e **Percorso**, produce il **list** aggiornato degli assegnamenti dopo l'esecuzione del comando. Inserito subito dopo il parametro **Device**, ne provoca l'eliminazione dalla lista degli assegnamenti (come se si adoperasse **Assign Device:**).

ASSIGN device: percorso LIST EXISTS REMOVE

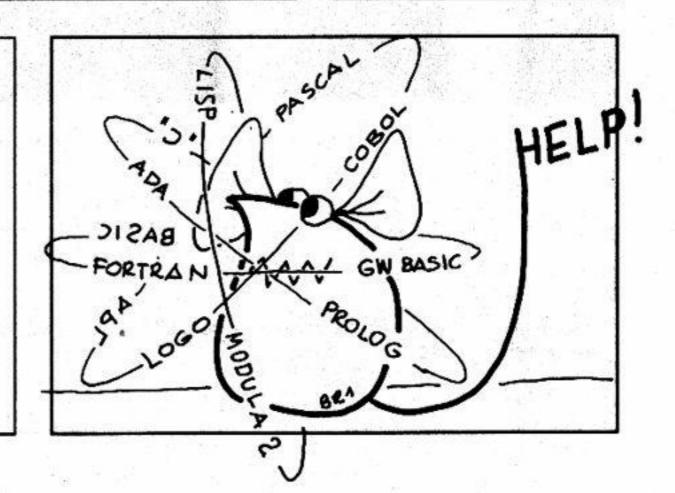
Device. Nome della device logica da definire. Deve obbligatoriamente concludersi con il simbolo **due punti**(:). Può fare riferimento a nomi di directory già esistenti, come pure a nuove unità logiche di nome facoltativo. Se non è seguita dal parametro **Percorso**, rimuove l'assegnazione logica legata al suo nome.

Exist. Opzionale. Adoperato dopo il parametro **Device**, ne visualizza l'attuale assegnazione, o ne segnala l'assenza. Utile soprattutto nei file batch, in quanto modifica la condizione **Warn** (= assegnazione non presente) **Not Warn** (= presente).

Remove. Opzionale. Svolge la stessa funzione di **Assign Device:** senza altri parametri, ma non rilascia la memoria impegnata. Praticamente,da non usarsi mai.

ERRORI PIU' FREQUENTI

Invalid device name
Con ogni probabilità, si è dimenticato di aggiungere i due punti al parametro *Device*.

Can't find....
Si è fornito, come parametro *Percorso*, un nome di directory o file inesistente. Controllare eventuali errori di digitazione, come pure incomplete descrizioni nel precisare eventuali subdirectory.

ESEMPI

Ecco alcune forme sintattiche di frequente impiego:
Assign Fonts: Df1:Fonts
Il sistema considererà la directory **Fonts**, che risulta posta nel secondo drive (notare, a tal proposito, la presenza di DF1:) come quella in cui ricercare la descrizione dei caratteri. Si noti che, in ogni caso, la ricerca verrà effettuata dapprima nella directory "corrente".

Assign Fonts:
Viene rimosso il device logico **Fonts:** (che, ovviamente, si suppone assegnato in precedenza), che risulterà così non più assegnato. Nel caso particolare, risulterà impossibile adoperare altri caratteri che non siano il Topaz (Font standard).

Assign X: Wb:System/Diskcopy
Dopo questa assegnazione, digitando **X:** (e Return) si manderà in esecuzione il file *Diskcopy* presente nella directory **System** del disco **Wb**. In questo caso, occorrerà aggiungere, nella linea di comando, i parametri legati al programma **Diskcopy** (per esempio **X: Df0: To Df1:**).

LA FUNZIONE ASSIGN

La funzione assegna ad un *Device logico* un percorso fisico, che può identificarsi tanto con una directory che con un singolo file.
I Device di default (**S:, C:, L:, LIBS:, FONTS:, DEVS:, SYS:**) vengono assegnati, al momento del boot, alle omonime directory del disco di lancio (se esistenti), con **Sys:** che identifica la directory principale (Root). A meno, ovviamente, di nuovi assegnamenti presenti nella Startup-sequence del boot disk.

ATTENZIONE A...

Non escudete assegnazioni di importanza vitale per il sistema. Se, per esempio, si assegna il device logico **Devs:** ad un'altra unità disco, e poi si vuole adoperare la stampante, ci si assicuri che nella directory **Devs** del disco in questione sia presente il file **Printer.device**, e la subdirectory **Printers**. In caso contrario, la stampa su carta risulterebbe inibita.

COME USARE LE TAVOLE

Le parole chiave del comando sono espresse in maiuscolo e vanno (eventualmente) adoperate così come sono. In minuscolo sono invece rappresentati i parametri che vanno ridefiniti dall'utente.

COPY

Origine. Specifica il file o la directory da copiare. E' consentito l'uso dei cosiddetti **Pattern Matching** (tra cui la ben nota coppia **#?**) per duplicare più files. Se in **Origine** non è specificato il percorso, viene assunta per default la directory **corrente**. Il parametro può essere omesso, nel qual caso (con l'obbligo di usare **TO**) verrano copiati **tutti** i files della directory corrente. Se **Origine** è il nome di una directory, la stessa verrà creata nel percorso **Destinazione** e tutti i files in essa contenuti saranno copiati.

Destinazione. Deve comprendere il percorso completo. Se non si intende modificare il nome del file da copiare, questo può essere omesso e verrà assunto lo stesso nome di **Origine.** Specificando solo un nome di file (senza percorso) diverso da quello origine, si provocherà la sua duplicazione all'interno della stessa directory.

Buffer. Opzionale. Con **XX** si indica la quantità di memoria da riservare come *buffer* per la copia. Il valore da assegnare ad *XX* è espresso in numero di buffer, ognuno dei quali assomma 512 byte. Il valore di default (200) va più che bene, per cui è una scelta quasi da ignorare.

All. Da aggiungere quando si intende copiare il contenuto di un disco o di una directory comprendendo **tutti** i files di **tutte** le eventuali subdirectory incluse.

Quiet. Opzionale. Elimina la descrizione delle varie fasi di copia, che di solito appaiono sul video.

COPY orig. destin. ALL QUIET BUF=xx CLONE DATE COM NOPRO

Clone. Opzionale. Copia i files mantenendo tutti i settaggi dell'originale (visibili con **List**): data, bit di protezione ed eventuali commenti.

Nopro. Opzionale. Non copia lo stato dei bit di protezione del file originale.

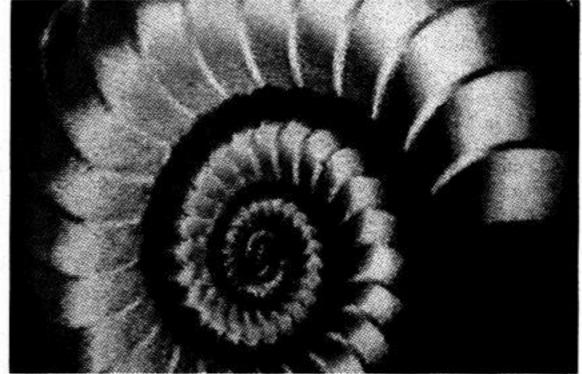
Date. Opzionale. Mantiene, nei settaggi della copia, solo la stessa data del file originale.

Com. Opzionale. Mantiene, nel file destinazione, l'eventuale commento associato al file origine.

LA FUNZIONE COPY

Copia uno o più files da una directory all'altra o da una periferica ad un'altra. Consente anche di duplicare un file nell'ambito della stessa directory, con nome ovviamente diverso. Nell'operazione di copia possono essere coinvolte anche **intere** directory, che vengono automaticamente create qualora non risultassero presenti nella periferica di destinazione (solo nella versione 1.3)(*)
In alternativa, la sintassi del comando può assumere la forma...
COPY FROM Origine TO Destinazione
Se si adopera **From**, è poi indispensabile adoperare anche **To**. Al contrario, l'uso del **To** tra **Origine** e **Destinazione** è consentito anche in assenza di **From**.

ATTENZIONE A...

Evitare involontarie perdite di file preesistenti, soprattutto nel caso di copie multiple: se un file **Origine** ha lo stesso nome di uno **Destinazione**, non si dimentichi che il secondo verrà sovrascritto **senza** alcun preavviso, perdendo definitivamente il suo precedente contenuto.

ESEMPI
Per chiarire l'uso del comando riportiamo alcuni esempi di largo utilizzo:

Copy df0:c ram:
Copia tutti i files contenuti nella directory **C** presenti nel disco in **df0:** nella **Ram disk**.

Copy df0: df1: All Quiet
Copia l'intero contenuto di **df0:** in **df1:**, comprese tutte le subdirectory del disco, senza mostrare su video l'andamento della copia.

Copy #?.info to ram:
Copia tutte le icone (file con suffisso **.info**) contenute nella directory corrente in **Ram Disk**.

Copy to Ram:
Copia tutti i files della directory corrente in **Ram Disk**.

ERRORE CLASSICO

Can't open....
Si è fornito un nome o (più frequentemente) un percorso scorretto per i parametri **Origine** oppure **Destinazione**.

N.B. - Le parole chiave del comando sono espresse in maiuscolo e vanno (eventualmente) adoperate così come sono. In minuscolo sono invece rappresentati i parametri che vanno ridefiniti dall'utente.

SEARCH

Nome. Può essere il nome di un file (compreso l'eventuale percorso completo) all'interno del quale limitare una ricerca, come pure il nome di una *directory* o di un *disco*. In questo caso, verranno scanditi tutti i files della directory (o disco), senza però accedere ad eventuali subdirectory.

All. Opzionale. Aggiungendo questo parametro quando **Nome** specifica una directory o disco, la ricerca verrà estesa anche a tutti i files presenti in ogni ordine di subdirectory.

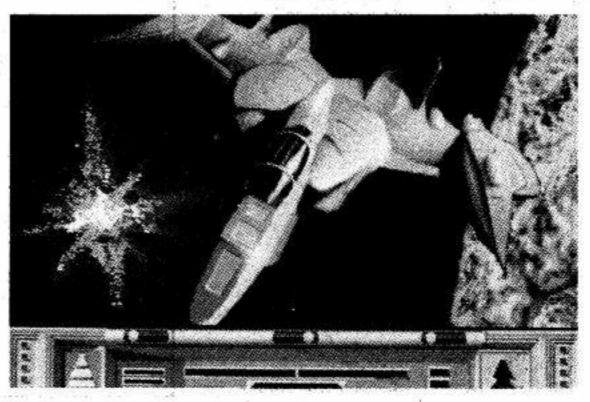
Quiet. Opzionale. Durante la ricerca, sullo schermo non viene visualizzato nulla. Quando il programma conclude la sua opera, viene fornito in output solo il nome del file che contiene la stringa. Non funziona (come ovvio) se la ricerca è limitata ad un solo file.

SEARCH nome stringa ALL NONUM QUIET QUICK FILE

Stringa. E' la stringa di caratteri da ricercare. Se contiene al suo interno degli spazi, deve essere racchiusa tra doppi apici.

Nonum. Opzionale. La visualizzazione della stringa trovata non è preceduta dal numero di riga.

Quick. Opzionale. Modifica lievemente l'output sullo schermo, senza alcun reale vantaggio. Pressochè inutile.

File. Opzionale. Se aggiunto, il parametro **Stringa** viene assunto come nome di file, e *Search* verifica se, nella directory (o disco) specificato dal parametro **Nome**, il file è presente. Impartito direttamente da Shell o Cli, si limita a visualizzare il nome eventualmente trovato. Più utile nell'ambito di un batch file (magari associato a Quiet), in quanto modifica la condizione Warn (Warn = file non trovato, Not Warn = file trovato).

N.B.: La sintassi prevede anche l'uso delle parole chiave **FROM** e **SEARCH**, per esempio...

Search From nomedir Search stringa

...che possono essere omesse senza alcun problema. Giusto per saperlo nel caso la si incontri, ma del tutto superflua nell'uso pratico.

ATTENZIONE A...

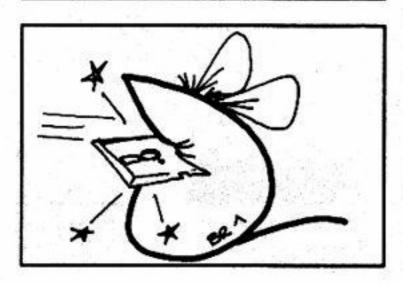
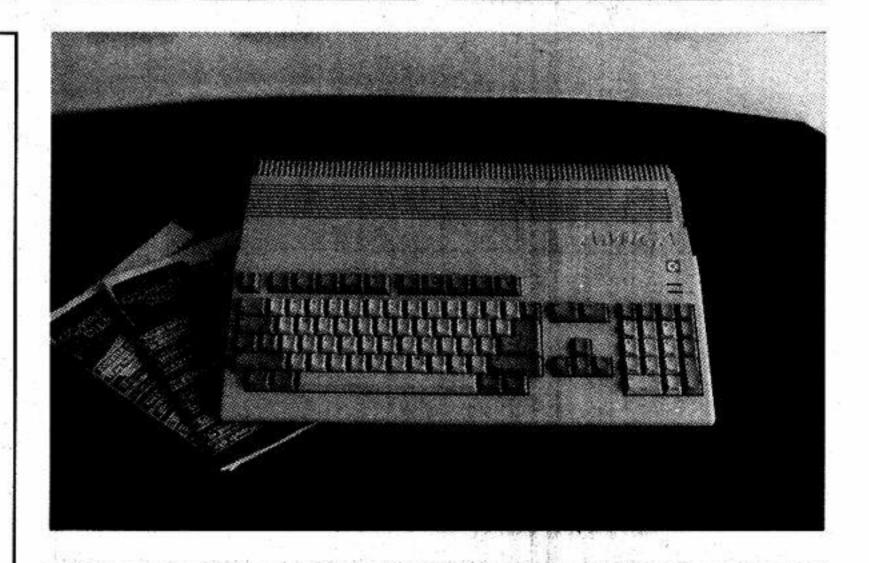
Durante l'esecuzione del comando, può capitare di veder apparire il messaggio **Warning: line xx too long**, dovuto al fatto che Search limita la sua ricerca ai primi 205 caratteri di ogni linea, considerando come fine linea (**EOL**) un Line Feed (**chr$(10)**).
La segnalazione indica che non si è potuta scandire tutta la riga, in quanto supera il limite concesso.

LA FUNZIONE SEARCH

Scandisce uno o più file di una directory o di un disco, verificando se al loro interno è presente una stringa fornita come parametro del comando. Se la ricorrenza indicata viene trovata, di norma viene visualizzata sullo schermo assieme al numero di riga del file, ed alla frase che la contiene. La ricerca è estesa tanto ai files testo che a quelli eseguibili, e non fa differenza tra caratteri in maiuscolo e in minuscolo.

ERRORE PIU' FREQUENTE

Bad arguments
A parte la possibilità di aver digitato erroneamente una delle parole chiave, l'errore può verificarsi (più subdolamente) anche quando il parametro **Stringa** è composto da più termini, e si è dimenticato di racchiudere il tutto tra **doppi apici** ("). In questo caso, per esempio in *Search df1: porca miseria,* il comando interpreterà il vocabolo *miseria* come una parola chiave, ovviamente non contemplata dalla sua sintassi. La forma corretta sarà dunque...
Search Df1: "porca miseria"

Could not get information...
Solito errore nella segnalazione del percorso per rintracciare il file o la directory del parametro **Nome**.

ESEMPI

Search ram: carattere
Cerca la stringa *carattere* in tutti files contenuti nella **Ram Disk**.

Search #?.txt "Gulp! Gasp!"
Cerca, nella directory corrente, la ricorrenza *Gulp! Gasp!* in tutti i files che terminano con suffisso **.txt**.

```
; demo file batch
Search >nil: Sys:c Run file
if warn
echo "Manca Run!"
quit
endif
echo "Run c'è"
```

Verifica che nella directory **C** del disco di boot sia presente il comando **Run**. In sua assenza interrompe l'esecuzione del batch con una segnalazione, altrimenti continua. La ridirezione **>nil:** serve per non far apparire il nome del file rintracciato.

DIR

Dir. Impartito da solo, lista il contenuto della directory corrente. Vengono mostrati dapprima i nomi di eventuali subdirectory presenti (senza il loro contenuto), poi quelli dei singoli files.

All. Parametro opzionale. Aggiungendolo, viene listato anche il contenuto di eventuali subdirectory, per ogni ordine di nidificazione esistente.

Files. Opzionale. Lista solo i nomi dei files, escludendo eventuali directory.

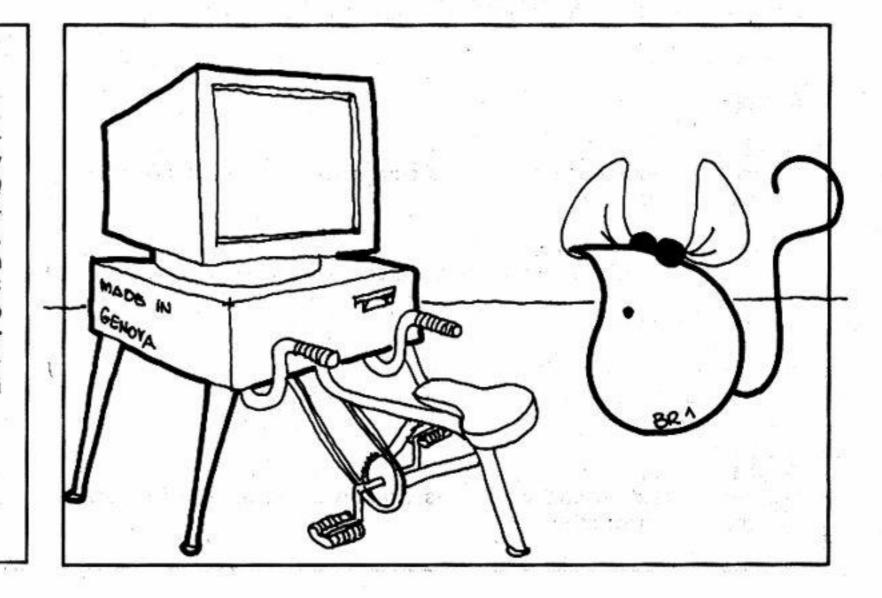
DIR nome ALL DIRS FILES INTER

Inter. Opzionale. Modo Interattivo. Viene visualizzato un nome alla volta seguito da un punto interrogativo; il computer resta in attesa che si impartisca, da tastiera, un comando, formato per lo più da una sola lettera e seguito da Return (o Enter). Con la pressione del solo Return si passa al successivo elemento della lista. Per uscire dall'esecuzione, inserire **Q** come risposta o premere il tasto **Ctrl** più la barra inversa (\).

Nome. Indica il nome della directory o del device logico da scandire. Nel caso di subdirectory, è necessario precisare l'intero percorso.

Dirs. Opzionale. Vengono visualizzati solo i nomi di directory, escludendo quelli dei files.

ERRORE PIU' FREQUENTE

...Not Found
Non si è specificato il corretto percorso per rintracciare la directory voluta. Se, per esempio, con **CD** si è impostata come directory corrente **Devs** del disco *Workbench*, impartendo **Dir C** si incorrerà in questo tipo di errore, in quanto C non è *visibile* da Devs. Il modo corretto sarà **Dir:C**, oppure **Dir Sys:C**, o ancora **Dir WB:C** (supponendo che il nome del disco sia **WB**).

COMANDI DEL MODO INTERATTIVO

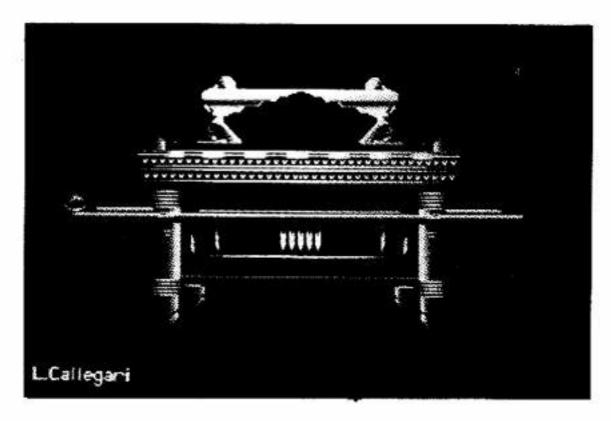
T
Visualizza il contenuto di un file, esattamente come se si fosse impartito un comando **Type**. Non funziona, come ovvio, per i nomi di directory. Premere un *tasto* qualunque per una pausa e *Back Space* per continuare. Per interrompere l'esecuzione premere **Ctrl+C**, che farà tornare al modo interattivo di Dir.

E
Consente di addentrarsi in una subdirectory, il cui contenuto sarà listato sempre in modo interattivo. Valido solo in risposta a nomi di directory.

B
Risale alla directory superiore (Parent), continuando la scansione interattiva.

Q
Interrompe l'esecuzione di Dir, tornando al prompt di Shell.

C
Permette l'esecuzione di un comando Dos senza uscire dal modo interattivo di *Dir*.
Viene visualizzato un prompt *Command?*, al quale bisogna rispondere con il nome del comando voluto (per esempio **CD**). Dopo l'esecuzione, la scansione interattiva ricomincia dall'elemento della lista successivo.

DEL
Cancella il file listato, senza chiedere conferma.
Una directory verrà cancellata solo se vuota.

ESEMPI

Dir Devs
Mostra il contenuto della directory Devs, purchè questa sia accessibile dalla directory corrente.

Dir C:
Lista il device logico **C**, ovvero la directory **C** del disco adoperato per lanciare il sistema (salvo diverse assegnazioni tramite **Assign**), indipendentemente dalla directory corrente.

Dir D#? Files
Mostra i soli files contenuti in tutte le directory il cui nome comincia con **D**.

Dir Df1: opt Al
Dir Df1: All Inter
Scandisce in modo interattivo tutti i files, directory ed eventuali subdirectory presenti nel secondo drive.

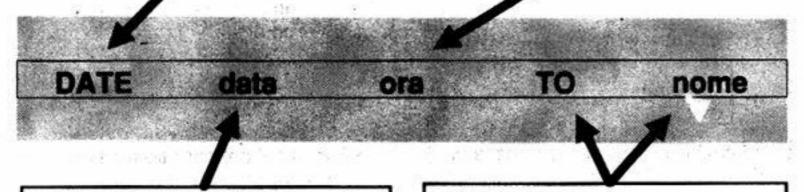
N.B. Le parole chiave **All**, **Dirs** e **Inter** possono essere sostituite, nell'ordine, da **OPT A**, **OPT D** e **OPT I**. I parametri *All* e *Inter* sono associabili, per cui è anche possibile adoperare **OPT AI**.

LA FUNZIONE DIR

Elenca il contenuto di un disco o di una directory, mostrando i nomi dei files ordinati alfabeticamente. Consente anche di interagire con i singoli files.

DATE

Date. Impartito da solo, visualizza la data e l'ora corrente, o meglio quella che il sistema *vede* come tale (prelevata dal disco di boot), nel caso non sia correttamente settata.

Ora. L'ora va regolata inserendola (anche in assenza del parametro Data) nel formato **HH:MM:SS**, compresi i simboli doppio punto tra le cifre, che vanno così interpretate: **HH** sono le ore, **MM** i minuti, **SS** i secondi. Questi ultimi possono anche essere omessi. Le ore sono da intendersi nella notazione 24 ore (per esempio 18 per le 6 pomeridiane), e, così come per minuti e secondi, è consentito introdurne il valore anche con una sola cifra (non è necessario anteporre lo zero).

Data. Adoperando questo parametro, si regola la data corrente. Il formato da adottare è...
GG-MMM-AA
...trattini compresi, con *GG* che indica il giorno, *MMM* il mese (prime tre lettere del nome **in inglese**), *AA* le ultime due cifre dell'anno. E' anche possibile adoperare per questo parametro i termini **Today** (oggi), **Tomorrow** (domani) e **Yesterday** (ieri), che modificheranno di conseguenza la data corrente. Analogamente, sempre in inglese, si può adoperare il giorno della settimana (per esempio *Saturday* per Sabato, ecc.).

Nome. Parametro opzionale. Può indicare il nome di un file o di una periferica verso cui *redirigere* la data attuale, peraltro automaticamente gestita dal sistema. Si presti attenzione se si adopera il nome di un file, in quanto questo, se già presente, verrà sovrascritto perdendo il suo contenuto precedente.
Per questa opzione, è obbligatorio inserire **TO**.

ESEMPI

Date 8-Oct-90 16:00

Date Wednesday
Imposta la data al prossimo mercoledì.
Date 1-Jan-00 0:0:0
Capodanno dell'anno 2000.

LA FUNZIONE DATE

Visualizza o modifica la data e l'ora di sistema. L'indicazione settata può poi essere salvata nell'eventuale Clock-tampone contenuto nelle espansioni di memoria o a corredo degli A2000 mediante il comando *Setclock Save*. Il settaggio in corso viene automaticamente associato a qualunque file venga manipolato.

ERRORE PIÙ FREQUENTE

Bad Args
Erronea digitazione del formato di data oppure ora. In particolare, si badi alla traduzione in inglese del mese (**Dec** per December, **May** per Maggio, ecc.).

ATTENZIONE A...

Se si intende mantenere la data associata ai vari files di un dischetto, si eviti il semplice comando Copy. In questi casi è preferibile ricorrere a *Diskcopy* (copia l'intero disco), che non modifica l'associazione della data, oppure sfruttare il parametro Date (anche Clone può andar bene) associato a Copy.

INSTALL

Install. Impartito da solo, visualizza la corretta sintassi da adottare per usare il comando stesso.

dfx: Specifica la periferica sulla quale Install deve agire (**Df0:**, **Df1:**, eccetera).

Check. Opzionale. Permette un controllo del bootblock, che non viene comunque modificato.

INSTALL DRIVE dfx: NOBOOT CHECK

Drive. Parametro facoltativo. Può essere omesso senza problemi.

Noboot. Opzionale. Aggiungendo questo parametro, si ottiene l'effetto opposto ad **Install**. L'eventuale bootblock memorizzato sul dischetto viene infatti cancellato, ed il disco non sarà più dotato di autostart.

ATTENZIONE A...

Il comando *Install*, ad eccezione dell'uso associato al parametro *Check*, opera indiscriminatamente sul blocco di start di un dischetto. Prima di adoperarlo su supporti magnetici contenenti software commerciale, occorre accertarsi che il bootblock non sia determinante ai fini del corretto funzionamento del software. Lo stesso dicasi per talune raccolte di utility, il cui menu di scelta è spesso direttamente gestito dal bootblock.

ERRORI PIÙ FREQUENTI

Disk is write protected
Il disco, come ovvio,deve essere abilitato alla scrittura.

No disk in unit...
Si commenta da sè.

LA FUNZIONE INSTALL

Rende autobootabile un dischetto, consentendo ad Amiga Dos di avviare la procedura di inizializzazione basandosi sulla startup-sequence del floppy così trattato. Dopo il boot, il dischetto adoperato per il lancio verrà assunto come device logico **Sys:** (che sta per Sistema).

MESSAGGI LEGATI A "CHECK"

No bootblock installed
Il disco non è dotato di autostart.

May not be standard V1.2/1.3 bootblock
Il disco è dotato di autostart, ma non quello standard Commodore. Non è escluso, in questo caso, che si tratti di un virus.

Appears to be normal V1.2/V1.3 bootblock
SEMBRA che il bootblock sia in regola (senza sbilanciarsi troppo: alcuni virus possono interferire sul giudizio).

ESEMPI

Install Df1:
Rende autobootabile il disco inserito nel secondo drive.

Install df0: Check
Opera un controllo sul bootblock del disco in Df0:

Install Drive Df1: Noboot
Rende non bootabile il disco inserito nell'unità interna Df0:.

N.B. - Le parole chiave del comando sono espresse in maiuscolo e vanno (eventualmente) adoperate così come sono. In minuscolo sono invece rappresentati i parametri che vanno ridefiniti dall'utente.

PATH

Path. Senza alcun parametro, visualizza l'elenco dei percorsi attivi al momento.

Add. Opzionale e facoltativo. Inserito dopo Path, precisa che l'elenco dei nomi che seguiranno (parametro **nomedirs**) sono da aggiungere in qualità di ulteriori percorsi di ricerca. Lo stesso effetto si ottiene digitando semplicemente *Path* seguito dai nomi da aggiungere alla lista.

Reset. Usato subito dopo *Path* senza altri parametri, ripristina i percorsi di default, ovvero la directory *corrente* e il device logico **C:**, eliminando eventuali altri percorsi in precedenza aggiunti. Aggiungendo dopo *Path Reset* il nome di nuove directory, queste verranno aggiunte alla lista dei path, e contemporaneamente verrano eliminati tutti gli altri percorsi, eccetto **C:** e della directory corrente.

PATH SHOW ADD nomedirs RESET

Show. Se aggiunto al comando, esegue esattamente lo stesso compito di Path, ma senza parametri. Praticamente da dimenticarne l'esistenza.

Nomedirs. Sono i nomi di *directory* o *device logici* nei quali ricercare i comandi impartiti da Dos. Con una singola istruzione *Path* (o **Path Add**) si possono aggiungere fino a 10 percorsi, separando con uno spazio i nomi delle directory. E' obbligatorio aggiungere il percorso completo dei nomi.

ATTENZIONE A...

Il comando Path è spesso utilizzato nella *startup-sequence* (vedi quella del disco Workbench). Se, però, si interrompe la procedura di boot con **Ctrl+D**, magari per abbreviare i tempi, può capitare che il comando Path non sia ancora stato eseguito. In questo caso (ci riferiamo all'esempio del disco Workbench), impartendo in *Cli* un comando *Format* si avrà una segnalazione di errore *Unknown Command*. Il file Format, infatti, è inserito nella directory *System* del disco di sistema, ed è necessario lanciarlo con **System/Format**. Se la procedura di start viene completata, un comando *Path System* presente nella startup-sequence fa sì che un semplice Format (impartito da Shell o Cli) funzioni.

LA FUNZIONE PATH

Mostra, oppure imposta, i percorsi di ricerca sfruttati dal sistema per rintracciare i comandi impartiti da **Shell** o **Cli** senza alcuna specifica di percorso. Per default vengono utilizzati la directory **corrente** e la directory **C**, o meglio il device logico **C:** (dopo lo start, identificabile con la directory **C** del disco adoperato per il boot). Se si è installata *Shell*, il percorso prevede anche una ricerca (prima di ogni altra) nella lista dei comandi resi residenti.

ESEMPI

Path Df1:c
Un comando sarà ricercato nella directory corrente, poi nella directory **C** assegnata come device logico (di solito quella del disco di boot), ed infine nella directory **C** del disco inserito nel drive **df1:**.
Path Sys: :System
Aggiunge al percorso di ricerca la directory principale del disco adoperato come boot, e la directory **System**, visibile dalla Root directory attuale (i due punti iniziali).

ERRORE PIÙ FREQUENTE

Unable to find...
Si è usato, come parametro **nomedirs**, una directory inesistente, magari per un errore di digitazione, oppure non se ne è precisato il percorso.

SORT

Origine. Nome del file-testo da riordinare, eventualmente completato dalla descrizione del suo percorso.

Destinazione. Nome del file destinazione, che conterrà il testo riordinato. Non è ammesso adoperare lo stesso nome del file origine. Se necessario, va anche qui fornito il suo percorso.

Colstart XX. Aggiungendo questa parola chiave, ai fini dell'ordinamento verrà preso in considerazione il carattere numero **xx** di ogni riga. Se il Sort trova uguali i caratteri xx di più righe, le riordina prendendo in considerazione il primo di ognuna di esse.

SORT FROM origine TO destinazione COLSTART xx

LA FUNZIONE SORT

Ridispone in ordine alfabetico il contenuto di un file Ascii, ovvero di solo testo, composto da una serie di righe. Ogni riga costituisce un elemento per il sort, per cui si può affermare che viene eseguito un riordinamento per righe, utile soprattutto in presenza di **liste** di dati. Una riga, per AmigaDos, è un insieme di caratteri che si concludono con un **Line Feed** (il codice Ascii **10** che viene per esempio inserito ad ogni pressione del Return quando si adopera l'editor **ED**). Sort non fa distinzione tra caratteri in maiuscolo e caratteri in minuscolo.

From, To. Parametri facoltativi. Possono entrambi essere omessi senza alcuna conseguenza. Qualora si utilizzi la keyword **FROM**, diventa però indispensabile adoperare anche la voce **TO**.

ATTENZIONE A...

Non utilizzare come file destinazione il nome di un file già esistente, che verrebbe sovrascitto senza alcun preavviso, perdendone irrimediabilmente il suo contenuto. Il comando Sort trova indicazione nel riordino di files non troppo lunghi, vuoi perchè non particolarmente veloce, vuoi perchè non può agire su files che contengano più di (circa) 200 righe. In questo caso, infatti, la memoria di lavoro assegnata per default al task attivo (4000 byte) non è più sufficiente, e si rende indispensabile incrementarla tramite il comando **Stack** (per esempio Stack 20000) in rapporto alle dimensioni del file, e procedendo per tentativi ed errori (Guru). Da evitare.

ESEMPIO

Si consideri un file Ascii così strutturato, di nome *Test*:

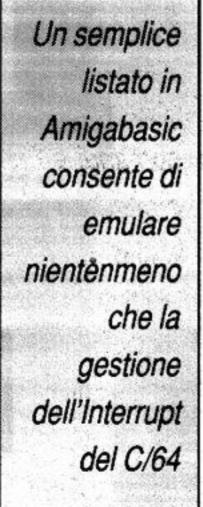
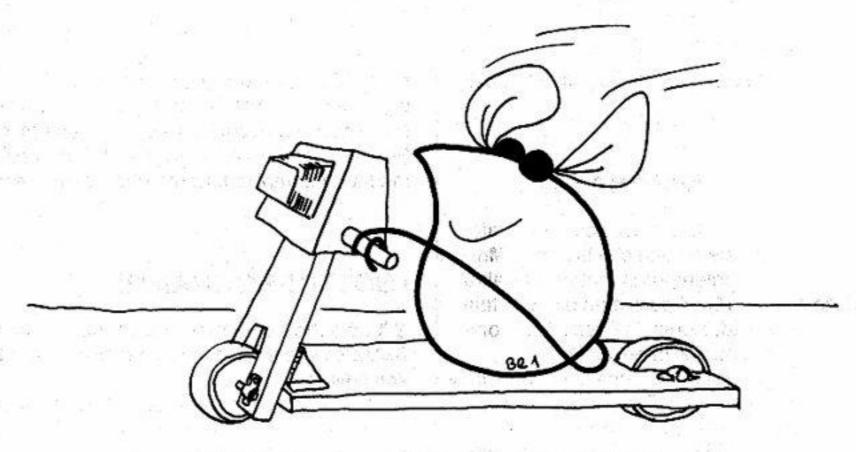
Antonio Brambilla
000123 Sigla
Superpippo
Gianni Settimio

Dopo un comando...
SORT test ord1
...si avrà, nella directory corrente, un file di nome *Ord1* con il seguente contenuto:

000123 Sigla
Antonio Brambilla
Gianni Settimio
Superpippo

Con il comando...
SORT test ord2 COLSTART 9
il file di nome *Ord2* assumerà invece questa connotazione:

Antonio Brambilla
Gianni Settimio
Superpippo
000123 Sigla

Si noti come l'ultima riga, che si suppone contenente **due** spazi tra il numero *00123* ed i rimanenti caratteri, viene riordinata in base alla **S**, nono carattere in quanto nella procedura vanno considerati anche gli spazi.

UNA CORSA A BORDO DEL JOY

Un listato piuttosto breve, ideale per chi inizia con AmigaBasic, decisaemente utile per capire come agiscono alcune istruzioni del potente linguaggio interprete

di Davide Martinenghi

Il programma descritto in queste pagine lavora in AmigaBasic e propone un algoritmo di gioco, tipico esempio di come far lavorare un computer in tempo (quasi) reale, riferendosi ai famosi games sportivi di tipo olimpionico. Vediamo come eseguire quanto detto nel migliore dei modi.

GLI INDICATORI

Il programma, realizzato in maniera piuttosto grezza (per essere modificato a piacere dai lettori mantenendone solo lo scheletro principale), è ambientato nello svolgimento di una gara di corsa.

Viene formulata innanzitutto una richiesta sui metri che volete percorrere, in una misura compresa tra 50 e 10000.

Terminata questa fase, tre Beep vi prepareranno alla partenza; in seguito dovrete martoriare il vostro joystick con violenti (ma veloci) colpi alternati a destra e a sinistra, che immediatamente faranno salire gli indicatori di potenza e di metri (quello del tempo va da solo).

La scelta di un indicatore di potenza e non di velocità espressa in km/è è stata dettata dal fatto che il suo calcolo avrebbe fatto perdere tempo prezioso all'algoritmo, che, essendo in Basic, deve essere ridotto al minimo.

Grazie alla struttura del programma, invece, calcolare la potenza risulta più comodo.

Quando i metri percorsi raggiungeranno il valore richiesto all'inizio, la gara sarà terminata ed un display mostrerà i risultati, tra cui la velocità media.

Se ottenete risultati impossibili (100 metri in 5 secondi, velocità media di 72 km/h...) vuol dire che è correntemente in funzione il **comando ff** (presente nella versione 1.3 di WorkBench); in questo caso non dovrete fare altro che togliere l'apostrofo davanti alle **Rem** (ove indicato nel programma) e cancellare le righe precedenti (o inserirvi un apostrofo-Rem): in questo modo dovrete esser davvero bravi per riuscire a percorrere 100 metri in 9.8 secondi senza utilizzare steroidi o droghe simili.

Bisogna però ammettere che, utilizzando il comando **ff**, l'algoritmo risulta molto più preciso, grazie alla sua velocità, ed il valore potenza

viene scandito in un tempo molto simile a quello reale.

IL TEMPO IN AMIGABASIC

Due routines formano il nocciolo del listato: quella principale ed una che lavora in **MultITasking**, cioè contemporaneamente all'altra (se avete avuto un C/64 possiamo dire che tale procedura è la "traduzione", in Basic, della complessa gestione l.m. dell'Interrupt 64).

Come attivare una routine che lavori nel modo descritto è molto semplice: il comando...

ON TIMER (.1) GOSUB Label

...soddisferà le vostre esigenze, in quanto provvede a richiamare ogni 0.1 secondi (cioè ogni decimo di secondo) la routine da voi specificata (nel nostro caso, la routine **Tempo**) , sempre che poi non dimentichiate di attivare il tutto con il comando **Timer On**.

La routine principale analizza, innanzitutto, lo stato del joystick, mediante il comando **Stick (2)**: in particolare vuole che lo stato sia **1** oppure **-1** (vedi assegnazione **joy = -joy**). In questo caso, cioè se avete mosso il joystick nella direzione richiesta, verrà incrementata la variabile **C**, che contiene la potenza, di un valore pari a **Inc**, il quale, a sua volta, verrà posto a **3/C**: tutto questo si traduce nell'ottenere un forte incremento della potenza quando questa è bassa, mentre un incremento debole se questa è già alta (corrispondenza inversa); ciò comporta che ad un certo punto l'incremento sarà talmente basso che sarà impossibile aumentare la potenza, considerando i limiti tecnici del joystick ed i limiti umani.

Poi si passerà ad incrementare i metri di **C/6**, e questo avverrà non solo quando muovete il joystick, ma anche negli altri casi, creando il realismo a cui prima accennavamo, facendo cioè avanzare l'ipotetico omino di una quantità di metri strettamente correlati alla vostra velocità.

MULTITASKING

Si stampano quindi i tre indicatori e la routine ricomincia.

Ma perchè mai, vi starete chiedendo, la potenza tenderebbe a non andare oltre un certo limite, ma (anzi) a diminuire dal momento che non c'è niente che...?

Ed è qui che entra in gioco la routine che lavora in MultiTasking.

Ebbene è proprio merito suo se la potenza tende a diminuire di 0.5 ogni decimo di secondo ed il tempo, invece, ad aumentare (**t = t +0 .1: c = c -0.5**). Nella routine principale una **If** verifica se i metri percorsi hanno raggiunto la misura specificata ed assegna il controllo alla routine **Stampe** che provvede a visualizzare i risultati mediante semplici formule che potete vedere nel listato e che non necessitano di spiegazioni.

I SOTTOPROGRAMMI

Due sottoprogrammi concorrono ad aumentare l'interesse per il listato: si tratta di **Getkey** e **Ninput.**

Il primo simula un Input controllato (di vecchia memoria) ed è molto utile per verificare la pressione di un tasto: **Getkey a$, b$, c$** significa che il computer stamperà **a$**, attenderà la pressione di un carattere tra quelli contenuti in **b$** e porrà tale carattere in **c$** (si noti che se **b$ = ""** tutti i caratteri saranno accettati).

Il secondo non è da meno, in quanto consente di effettuare l'input di un valore numerico intero positivo compreso tra un minimo ed un massimo, quindi...

NInput MIn%, Max%, A$, A

...significa che si stamperà **a$** e verrà effettuato l'input numerico di un valore compreso tra **mln%** e **max%** e verrà posto in **a**.

SUGGERIMENTI

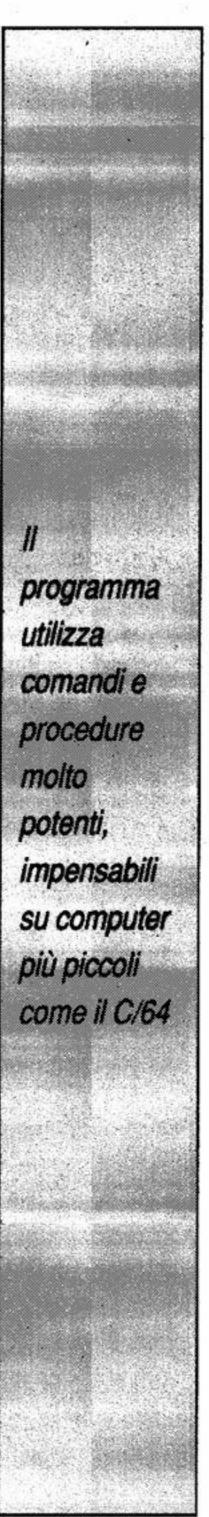
Il programma è totalmente privo di grafica; tocca a voi implementare l'utilizzo di sprites e disegni, nonchè di effetti sonori e musica.

Consigliamo, inoltre, di porre l'indicatore di potenza sotto forma di istogramma, magari usando **Line ....., bf**.

Potreste inoltre rimpiazzare i **Beep** iniziali con la voce di uno starter o aggiungere altre gare e, magari, tenere in memoria dei records e assegnare un punteggio in base al risultato ottenuto.

Nel realizzare migliorie, però, state attenti a non appesantire troppo il listato.

COPIALO PER TELEFONO

Anche i listati presenti in queste pagine possono esser tirati giù per mezzo del modem; se, ovviamente, ne possedete uno.
La procedura per collegarsi con la nostra banca dati (attiva 24 ore su 24) è riportata su altra parte della rivista.
Chi non possiede il modem (che aspettate a procurarvelo?) può tuttavia richiedere il dischetto, contenente il software, presso il nostro servizio arretrati.

'Destra & Sinistra - 1990
'Mini-game istruttivo per Amiga, di Davide Martinenghi

```
Corsa:

CLS: NINPUT 50, 10000, "Lunghezza percorso (50 - 10000)? ", met

CLS: COLOR 3, 0: PRINT: PRINT " Metri": PRINT: PRINT " Tempo"

PRINT: PRINT " Potenza": t = 0: m = 0
Start:
COLOR 1, 0: ON TIMER (.1) GOSUB Tempo

FOR j = 1 TO 3: FOR k = 1 TO 1000: NEXT k: BEEP: BEEP: NEXT j

TIMER ON: joy = 1: c = 1
inc = 3: REM mettere l'apostrofo!!!
'inc = 2: REM togliere l'apostrofo!!!
Routine:
IF STICK (2) = joy THEN joy = -joy: c = c+inc

IF c > = 1 THEN inc = 3 / c: ELSE inc = 3: REM mettere l'apostrofo!!!

'IF c > = 1 THEN inc = 2 / c: ELSE inc = 2: REM togliere l'apostrofo!!!

m = m+c / 6: REM mettere l'apostrofo!!!
'm = m+c / 9: REM togliere l'apostrofo!!!
IF m > = met THEN TIMER OFF: GOTO Stampe
LOCATE 2, 25: PRINT USING "#####.##"; m
LOCATE 4, 25: PRINT USING "#####.##"; t
LOCATE 6, 25: PRINT USING "#####.##"; c
GOTO Routine
Tempo:
t = t+.1: c = c-.5: IF c < 0 THEN c = 0
RETURN
Stampe:
CLS: PRINT: PRINT "Tempo impiegato "; :
PRINT USING "#####.##"; t
PRINT "Metri percorsi "; :

PRINT USING "#####.##"; met: v = (met / 1000) / (t / 3600)

PRINT "Velocità media "; :
PRINT USING "#####.##"; v; : PRINT " Km / h"
PRINT: PRINT: PRINT
GETKEY "Un'altra gara?", "sn", a$
IF a$ = "s" THEN Corsa
END
SUB GETKEY (a$, c$, g$) STATIC
PRINT a$;
GetKeyUno:
g$ = INKEY$
IF g$ = "" THEN GetKeyUno

IF c$ < > "" THEN IF INSTR (c$, g$) = 0 THEN BEEP: BEEP: GOTO GetKeyUno

END SUB
SUB NINPUT (min%, max%, a$, g) STATIC

IF min% < 0 OR max% < 0 OR INT (min%) < > min% OR INT (max%) < > max% THEN ERROR 5

PRINT a$; : f$ = ""
NInputUno:
inst = 0: g$ = "": WHILE g$ = "": g$ = INKEY$: WEND
IF g$ = CHR$ (8) THEN inst = 1: GOTO NInputDue
IF g$ = CHR$ (13) THEN NInputTre

IF ASC (g$) < 48 OR ASC (g$) > 57 THEN BEEP: GOTO NInputUno

NInputDue:

IF inst AND LEN (f$) < 1 THEN BEEP: GOTO NInputUno

IF inst THEN f$ = LEFT$ (f$, LEN (f$)-1): PRINT g$; : GOTO NInputUno

IF LEN (f$) > = LEN (STR$ (max%))-1 THEN BEEP: GOTO NInputUno

f$ = f$+g$: PRINT g$; : GOTO NInputUno
NInputTre:
g = VAL (f$)

IF g < min% OR g > max% THEN BEEP: GOTO NInputUno

PRINT
END SUB
```

POSTAMIGA

(A cura di Domenico Pavone)

LIBRARY NOT FOUND

Ho un problema che mi sta facendo impazzire: ogni volta che copio un listato basic che adopera le librerie, mi scontro sempre con una marea di errori "File not Found", che l'editor mi segnala riguardano il comando Library. Ho seguito i consigli che qualche volta avete dato al riguardo, ma senza esito.

(Giuseppe Bonocore - Milano)

In effetti, ed in modo particolare se si dispone di un solo drive, occorre prestare molta attenzione al percorso da associare al file con suffisso **.bmap** cui il comando **Library** fa riferimento. Si ricorda, a tal proposito, che nella directory BasicDemos del disco Extras 1.3 ed 1.2 sono già disponibili i files **Dos.bmap, Exec.bmap** e **Graphics.bmap**, sufficienti per la maggior parte delle applicazioni.

Schematicamente, ecco dunque un decalogo ("pentalogo" sarebbe forse più corretto) per rendere il file "visibile" al programma che deve gestirlo, tenendo presente che si può ricorrere facoltativamente ad una (ed una sola!) delle tecniche descritte:

1) Copiare il file .bmap nella directory **Libs** del disco adoperato per lo start al sistema (magari utilizzando una utility tipo *Dir Master, Climate,* ecc. per rendere il tutto più facile). Al comando Library, in questo caso, può essere associato il semplice nome del file (p.es. **Library "Dos.library"**).

2) Copiare il file .bmap nella stessa directory ove è memorizzato il programma, assegnandola poi come directory corrente del Basic (comando **Chdir "Nomedir"** impartito in modo diretto, o inserendolo nel listato *prima* del comando Library). Anche in questo caso, dopo Library, è sufficiente il solo nome della libreria.

3) Copiare il file .bmap (p. es.) in Ram Disk e modificare l'istruzione Library del listato trasformandola (p. es.) in Library "**Ram:Dos.library**".

4) Rinominare il disco Extras 1.3 in modo da eliminare lo spazio, magari adoperando solo Extras per il nuovo nome, e impostare per l'istruzione Library nel listato la forma Library "**Extras:BasicDemos/Dos.library**".

5) Chi possiede un secondo drive (gli altri si affrettino...), anche senza rinominare il disco Extras, possono utilizzare la tecnica del quarto "comandamento" adoperando nel programma la forma Library "Df1:BasicDemos/Dos.library". Come ovvio, si presuppone che il dischetto Extras sia già stato inserito nella seconda unità a dischi.

QUESTIONE DI STILE

Vorei sapere come si può, in AmigaBasic, produrre testi con caratteristiche particolari come fanno la maggior parte degli editor, Notepad compreso. Mi riferisco a neretto (Bold), sottolineato, e stile italico.

(Fabio Scandurra - Napoli)

Il cosiddetto "stile" di scrittura non è direttamente implementabile con AmigaBasic, tuttavia si può aggirare l'ostacolo ricorrendo alle funzioni della libreria grafica.

In pratica, occorre anzitutto trovare un valido sostituto al comando Print, inefficace per questo scopo.

L'equivalente funzione della **Graphic Library** si chiama Text, e va adoperata da Basic con la seguente sintassi:

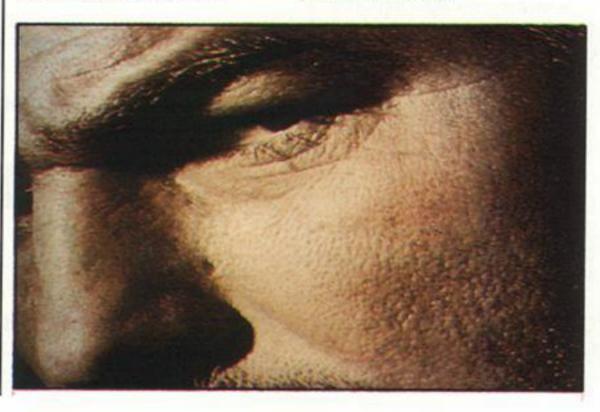
CALL Text& (Rast&, Indir&, Lungh&)

Le variabili-parametro racchiuse tra parentesi vanno così interpretate:

Rast& - Puntatore alla Raster Port della finestra corrente. In pratica, corrisponde alla funzione basic Window(8).

Indir& - Indirizzo di memoria ove è memorizzato il testo da stampare. Si tratta, molto semplicemente, di assegnare il testo ad una variabile stringa, il cui indirizzo è poi facilmente ricavabile tramite la funzione basic SADD(stringa$).

Lungh& - La lunghezza del testo, equivalente in basic a LEN(stringa$).

Si badi che la funzione Text non inserisce un ritorno carrello alla fine della stampa su video, per cui sarà opportuno un normale Print dopo la visualizzazione.

Prima di passare ad un esempio concreto, esaminiamo anche la sintassi di un'altra funzione della libreria grafica, quella che più specificamente ci consentirà di adoperare i diversi stili di scrittura richiamandola prima della già esaminata Text:

CALL SetSoftStyle& (Rast&, Stile&, Flag%)

La variabile **Rast&** assume lo stesso significato (e valore) già visto per la funzione Text, mentre **Stile&** è proprio lo "switch" che ci interessa per settare lo stile di scrittura.

Il valore di questa variabile va così interpretato:
0 = Normale.
1 = Sottolineato (underline).
2 = Neretto (boldface).
4 = Italico.

Per disporre di più settaggi contemporaneamente, basterà sommare i relativi valori (p. es. Neretto + sottolineato = 3).

Il terzo parametro, **Flag%**, serve per segnalare quali stili sono consentiti, in quanto alcuni tipi di Font possono non risultare compatibili con uno o più di essi.

Assegnando **255**, diremo al sistema che gli stili sono tutti possibili, dando per scontato che non si adopererà un set di caratteri alternativo (il **Topaz**, quello di default, accetta tutto).

Vedremo meglio tra breve come utilizzare il parametro.

A questo punto, disponiamo di tutte le cognizioni necessarie per tentare un esperimento.

Si copi, dunque, il breve listato sottostante, che necesita ovviamente della presenza del file **Graphics.bmap**, rintracciabile nella directory BasicDemos del disco Extras (vedi anche precedente risposta).

```
LIBRARY "graphics.library"
testo$="PROVA di testo"
lung&=LEN(testo$)
Indir&=SADD(testo$)
Rast&=WINDOW(8)
'------------------------------
FOR x&=7 TO 0 STEP -1
CALL SetSoftStyle& (Rast&, x&, 255)
CALL Text& (Rast&, Indir&, lung&)
PRINT: PRINT: NEXT
'------------------------------
LIBRARY CLOSE: END
```

Mandato in esecuzione, il programma mostrerà tutti gli stili possibili, tanto singolarmente che associati.

Il ciclo For...Next è percorso da 7 a 0 (e non da 0 a 7) in quanto l'ultimo stile impostato continua ad interagire con l'output del basic.

Con 0 come ultimo valore del loop si ripristina lo stile di default, risparmiando una ulteriore istruzione Call prima di concludere il programma.

Per quanto riguarda il parametro **Flag%**, nel caso si intenda adoperare un Font differente, sorge un unico dilemma: come sapere quale stile è eventualmente non impostabile, e quale valore assegnare di conseguenza alla variabile?

La risposta è molto semplice: lasciar fare tutto ad Amiga.

Il che, tradotto in relative istruzioni, significa richiamare (prima di impostare uno stile) un'altra funzione di libreria, **AskSoftStyle**, che restituisce un valore da passare poi come parametro Flag% a SetSoftStyle.

Ecco dunque come andrebbe "canonicamente" riscritta la routine precedente, anche se nel caso specifico sortirà esattamente lo stesso effetto.

```
LIBRARY "graphics.library"
DECLARE FUNCTION AskSoftStyle&() LIBRARY
testo$="PROVA di testo"
lung&=LEN(testo$)
Indir&=SADD(testo$)
Rast&=WINDOW(8)
'------------------------------
Flag%=AskSoftStyle&(Rast&)
FOR x&=7 TO 0 STEP -1
CALL SetSoftStyle& (Rast&, x&, Flag%)
CALL Text& (Rast&, Indir&, lung&)
PRINT: PRINT: NEXT
'------------------------------
LIBRARY CLOSE: END
```

ATTESE, LUNGHE ATTESE

Ho risistemato una copia del disco Workbench come disco lavoro, inserendovi tutti i files che mi interessavano.

Anche se tutto funziona come volevo, quando provo a dare List oppure Dir sembra che Amiga si sia... addormentata, impiegando un sacco di tempo per mostrare tutto il contenuto del disco. La stessa cosa, poi, succede anche in Workbench, con le icone che appaiono ad intervalli di un secolo circa...

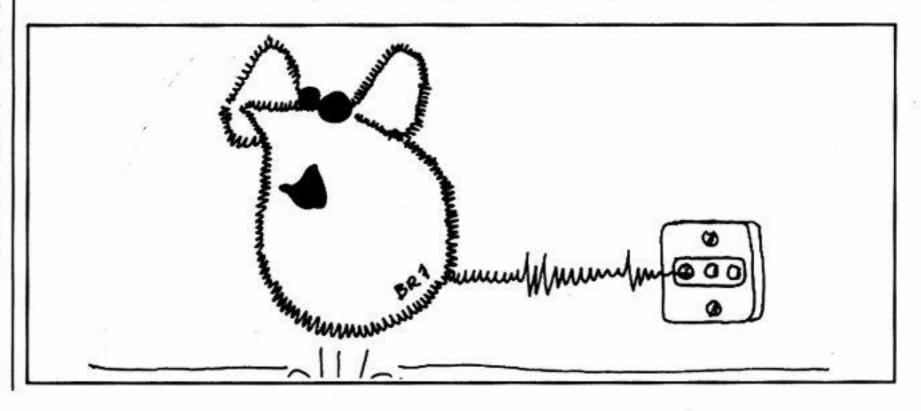
(Marco Piccini - Roma)

Beh, non c'è poi da meravigliarsi più di tanto.

Amiga Dos, infatti, ha una particolare gestione della directory di un dischetto, che non memorizza tutti i dati in uno specifico raggruppamento di blocchi.

Senza entrare nei dettagli (estremamente complicati), l'unico accesso "a colpo sicuro" della testina del drive è quello rivolto alla **traccia 40**, che contiene alcune informazioni di base sulla Root directory (quella principale).

Per il resto, le informazioni su files e directory varie sono memorizzate in ordine sparso (ma logico... almeno per Amiga) nei meandri del Floppy.

E' certamente un sistema meno veloce dei Dos che mantengono in un'unica area del disco tutte le informazioni che lo riguardano, ma che, di contro, consente di non perdere irrimediabilmente tutti i dati se, per malaugurato caso, quella particolare sezione venisse danneggiata.

Non a caso proliferano programmi come **DiskDoctor, Disk Save, Lazzaro,** eccetera.

Quando, però, si eseguono molte operazioni di cancellazione e riscrittura di files, il loro contenuto, e quindi anche gli Header Block che contengono le informazioni necessarie a Dir oppure List, è ancora più "sparpagliato" del normale.

Il che obbliga il sistema ad un continuo avanti-indietro della testina del drive, "inseguendo" i blocchi significativi ai fini della visualizzazione della directory.

Una maggiore velocità di scansione è quindi possibile solo se i files, o ancora meglio i vari blocchi di informazione, sono memorizzati in settori **contigui** del dischetto.

Dal punto di vista pratico, un certo incremento della velocità di scansione della directory, ma anche di caricamento dei programmi, si può ottenere semplicemente copiando l'intero disco file per file, e non ricorrendo a DiskCopy o similari.

L'optimum, però, lo si raggiunge adoperando appositi programmi, per lo più appartenenti al Pubblico Dominio.

Per non restare sul vago, un esempio tipico è **Disk Arranger,** che però opera solo su due drive, e che a tutt'oggi risulta il più efficiente.

A patto, però, che il disco da riordinare non contenga più di circa 150-200 files, o troppe nidificazioni di subdirectory, e che lo si lanci senza Fast Memory attiva.

Un altro programma di notevoli capacità è **Bad**, che supera i limiti prima visti da Disk Arranger (ma in compenso meno "speedevole"), e che consente una diversa ottimizzazione in base alla presunta utilizzazione del disco in ambiente Workbench oppure Cli.

Punto di forza di Bad, la capacità di operare anche sugli Hard Disk.

Buoni risultati si ottengono anche con il famoso **Xcopy**, certo più noto per le sue doti copierecce, ma che implementa una meno nota opzione *Optimize*.

Di recente apparizione, infine, un ulteriore "arrangia-dischi", a nome **Optimizer**, che, mentre svolge il suo compito, visualizza sullo schermo tanto la vecchia che la nuova organizzazione del disco

STAMPA DI QUALITA'

Posseggo, a dotazione del mio A-500, una stampante Commodore 1550-C che stampa benissimo sia in modalità grafica che in scrittura Draft o NLQ, a patto però che adoperi il Setup della stampante stessa.
Ho provato, infatti, ad usare questo breve programma basic..Dim Lett$ (26): A% = 97: B% =0

```
For X% =1 To 26
Lett$ (x%) = Chr$ (a% +
B%)
B% = B% + 1
Next X%
Lprint Chr$ (27);......
Open "lpt1:" For Output
As #1
For X%=1 To 26
Print #1, Lett$ (x%); " ";
Next X%
Close #1: End
```

...con, al posto dei puntini dopo Lprint, i codici indicati dal manuale per stampare (p. es.) in NLQ, ma l'output su carta mi risulta sempre e solo in Draft Mode, anche se tento di inviare i codici per il colore, il condensato, ecc.
(Firma non leggibile)

Il problema non riguarda la stampante, bensì AmigaBasic. L'argomento, pur con sfumature diverse, è stato affrontato dalla nostra rivista circa un anno fa, a proposito della stampa espansa.

L'**NLQ**, o *Near Letter Quality*, nelle stampanti ad emulazione Epson richiede (salvo diversa indicazione del manuale) p. es. che si invii un codice di **Escape** seguito da **"X"** e da **Chr$ (1)** per abilitarlo oppure **Chr$ (0)** per disabilitarlo. Nel listato inviato dal lettore tale sequenza viene inviata tramite l'istruzione **Lprint**, che però... non invia proprio niente.

Non in generale, si intende, ma solo quando si tratta di codici speciali, come appunto il Chr$ (27), cioè il codice Ascii di Escape.

Lo stesso dicasi per il device **Lpt1**, tra l'altro decisamente poco utile, data la presenza dei più comodi e versatili **Par:** e **Prt:** (abbreviazioni di **par**allel port e **pr**in**t**er).

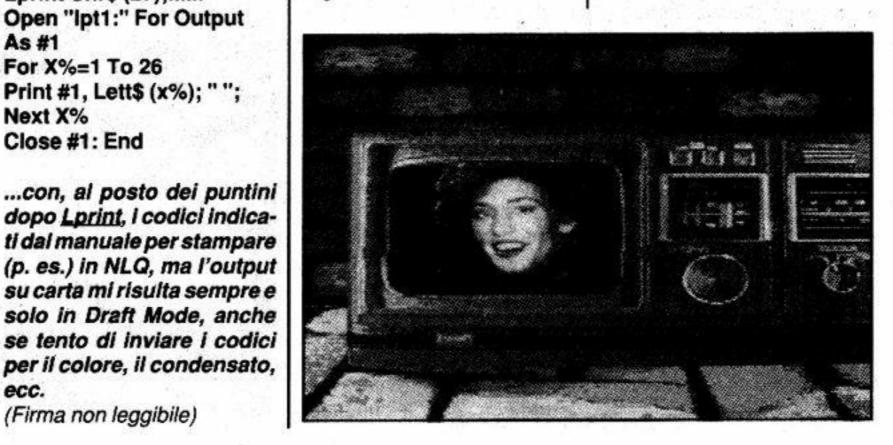
Ed è proprio con il device Par: che si può inviare la sequenza, come del resto anche i caratteri da stampare.

In definitiva, per abilitare l'NLQ e disabilitarlo dopo la stampa, il listato in questione va così corretto:

```
Dim Lett$ (26): A% = 97: B%
= 0
For X% =1 To 26
Lett$ (x%) = Chr$ (a% + B%)
B% = B% + 1
Next X%
Open "par:" For Output As #1
Print #1, Chr$ (27); "x"; Chr$
(1)
For X% = 1 To 26
Print #1, Lett$ (x%); " ";
Next X%
Print #1, Chr$ (27); "x"; Chr$
(0)
Close #1: End
```

Una nota conclusiva, anche se non proprio in argomento: un listato come quello proposto dal lettore, pur se corretto e "forbito", non fa altro che stampare in minuscolo tutte le lettere dell'alfabeto.

A meno di particolari esigenze, lo steso scopo lo si può raggiungere con un più banale, ma decisamente meno ingombrante...

```
Open "par:" For Output As #1
Print #1, Chr$ (27); "x"; Chr$
(1)
For X% = 97 To 122
Print #1, Chr$ (x%); " ";
Next X%
Print #1, Chr$ (27); "x"; Chr$
(0)
Close #1: End
```

QUANTI AMIGA?

Vorrei cambiare il mio Amiga 500 con un 2000, ma non so se acquistarlo subito, oppure aspettare i nuovi 3000 di cui si parla in giro. E' vero che quelli attuali non saranno più compatibili con i modelli futuri?
(Massimo Segafredi - Roma)

Intanto, per attendere le sempre imminenti migliorie di un computer prima di acquistarlo, si rischia di invecchiare seza comprare un bel niente, visto il ritmo incalzante degli avvenimenti (e non solo per Amiga, si intende).

Comunque, "a naso", gli Amiga 3000 che (**pare**) faranno la loro comparsa negli States più o meno a Luglio (in Italia non ci è dato sapere quando) non sono da associare alla stessa fascia di acquisto dei 2000.

Si dovrebbe trattare, infatti, di macchine a 32 bit basate sul microprocessore 68030 (e 5 nuovi chip a sostegno dei Custom), con clock a 16 e 25 Mhz (accidenti!), fornite di coprocessore matematico 68882, 2 Megabyte di memoria in dotazione e la possibilità di gestirne fino ad un GIGAbyte.

Come ovvio non mancherà un bell'Hard Disk da 40 (o, a scelta, da 100) Mega già "servito" con il computer, e, come annunciato dalla Commodore, con un Dos 2.0 nuovo di zecca.

In pratica, non si tatterà solo di un miglioramento degli attuali 2000, ma piuttosto di macchine rivolte ad una fascia più alta di utenza, soprattutto con finalità professionali.

Il che significherà anche prezzi che, per gli Stati Uniti, sono sin d'ora preventivati dai 4 - 5 milioni in su a seconda delle configurazioni.

In Italia, a meno che non venga smentita una ormai radicata consuetudine, i prezzi di solito vengono anche maggiorati.

In ogni caso, il Dos 2.0 (assolutamente misterioso al momento in cui vengono scritte queste righe) verrà poi immesso sul mercato, dopo Settembre, anche per gli A2000 come kit da montare sulle macchine preesistenti.

Per quanto riguarda la compatibilità, è impossibile stabilire qualcosa a priori, comunque i possessori degli attuali 2000 possono dormire sonni tranquilli: la presenza sul mercato di una marea di 500, impedirà per ancora molto, molto tempo che sorgano problemi "verso il basso".

In prospettiva, poi, un 2000 può sempre essere modificato o espanso con facilità: qualche scheda, una manciata di Chip... Chi vivrà, vedrà.

BASIC E BBS

Scusatemi, ma sono un "amigo" alle prime armi. Vorrei sapere se è possibile adoperare il modem con un programma scritto in basic, e se in tal modo ci si può collegare alla vostra banca dati.
(Vincenzo Cutroneo - Palermo)

Non c'e proprio niente di cui scusarsi: nel tempo, siamo stati **tutti** alle prime armi, l'importante è non restarci...

Ma veniamo al quesito.

Il basic, in effetti, non è che sia proprio l'ideale per un programma di comunicazione, almeno se si cercano prestazioni particolarmente spinte.

Tuttavia, per collegarsi semplicemente ad una banca dati (come BBSystems) e leggere / scrivere qualche messaggio, o per dialogare via modem con un amico, non è poi così difficile elaborare un programmino basic che ci dia una mano.

Anzi, bastano solo poche righe di istruzioni per disporre di un miniterminal come quello che si può vedere pubblicato in queste pagine, struttura di base per eventuali sperimentazioni più sofisticate.

Per provarlo, non c'è che da copiare il listato e mandarlo in esecuzione, badando che, nel caso non si sia lanciato il sistema col disco Workbench, sia presente il file **Serial.Device** nella directory **Devs** del bootdisk.

Naturalmente, al computer deve essere collegato un modem **Hayes compatibile** (praticamente lo sono tutti) attraverso la porta seriale (non penserete che Amiga comunichi a voce?). L'unico problema che sorge con il basic, è quello della velocità.

Sfogliando il manuale, infatti, si può notare come sia estremamente facile accedere al modem, in quanto esiste un device apposito, **Com1:** (coi due punti finali!), manipolabile esattamente come qualunque altra periferica, a suon di **Print#** ed **Input$()**.

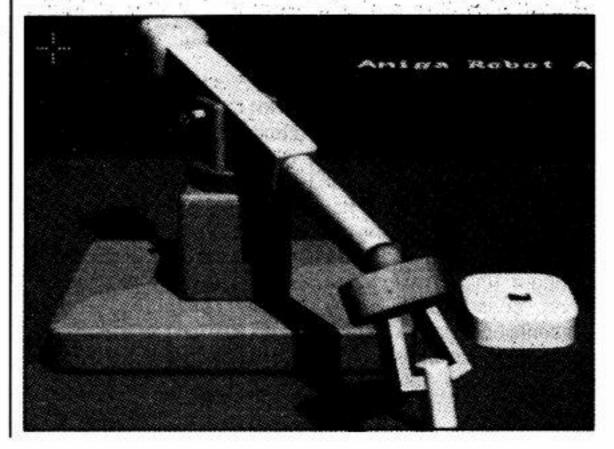
In pratica, occorre aprire un file che fa riferiferimento a Com1, settando nella stessa istruzione i parametri della trasmissione:

Open "Com1: baud, parità, bitdati, bitstop"

Senza entrare nel dettaglio, gli ultimi tre parametri (per le comuni connessioni) vanno impostati ad **N**, **8** e **1** (vedi listato), mentre a **baud** (la velocità espressa in bit per secondo) si potrà assegnare un valore di 300, 1200 e (teoricamente) 2400, a seconda delle capacità del modem.

Adoperando il basic, c'è però un limite. Quando, per esempio, si inviano dei dati ad una stampante, molto più lenta dei tempi di elaborazione del computer, quest'ultimo (generalizzando un po') attende che i caratteri siano stati stampati prima di inviarne altri, e lo stesso avviene con le unità a dischetti.

Con il modem, invece, tutto ciò che non è "catturato" all'istante, viene perso. Il che, considerando i tempi dell'interprete del basic, si traduce in una certa inaffidabilità (soprattutto in ricezione) a velocità che superano i 1200 baud.

A meno, ovviamente, di non utilizzare gli speciali **protocolli** di trasmissione studiati per supplire a questi limiti, ma è un argomento non esauribile in questa sede, e comunque di scarsa praticità

```
' MINITERMINAL HAYES COMPATIBILE
'
SCREEN 2, 640, 254, 1, 2
WINDOW 2, "M E N U", , 16, 2
MENU 1, 0, 1, "TELEFONO"
MENU 1, 1, 1, "TELEFONA"
MENU 1, 2, 1, "RITELEFONA"
MENU 1, 3, 1, "QUIT"
MENU 2, 0, 1, "ECHO"
MENU 2, 1, 1, "ON"
MENU 2, 2, 1, "OFF"
MENU 3, 0, 1, "": MENU 4, 0, 1, ""
WIDTH 80: ON MENU GOSUB scelta
MENU ON
OPEN "com1: 1200, N, 8, 1" AS #1
PRINT #1, "ATE0X3"
ON BREAK GOSUB ferma: BREAK ON
'----------- TERMINAL ------------
WHILE -1
 k$ = INKEY$: IF k$ > "" THEN
 PRINT #1, k$;
 END IF
 IF LOC (1) THEN PRINT INPUT$ (1, 1);
WEND
'--------- MENU GENERALE ---------
scelta:
id = MENU (0): item = MENU (1)
IF id = 1 THEN
 IF item = 1 THEN dial
 IF item = 2 THEN redial
 IF item = 3 THEN fine
END IF
IF id = 2 THEN
 IF item = 1 THEN PRINT #1, "ATE1"
 IF item = 2 THEN PRINT #1, "ATE0"
END IF
RETURN
'---- INPUT NUMERO TELEFONICO ----
dial:
INPUT"NUMERO DA CHIAMARE"; nr$
redial:
PRINT: PRINT "ATTENDI..."
PRINT #1, "ATDP"+nr$: RETURN
'------ INTERCETTA STOP ----------
ferma:
PRINT #1, CHR$ (3);: RETURN
'----- USCITA DAL PROGRAMMA ------
fine:
BREAK OFF: CLOSE 1: MENU RESET
WINDOW CLOSE 2: WINDOW CLOSE 3
SCREEN CLOSE 2: END
```

avendo a che fare con il basic.

Il programma proposto, quindi, è pronto per essere adoperato a 1200 baud, ma si può cambiare il parametro di Open come più si preferisce.

Si può infatti portarlo anche a 2400 baud, dato che non sono previste opzioni di invio / ricezione files, perdendo tutt'al più la formattazione delle schermate iniziali delle varie BBS.

Per migliorarlo, si potrebbe con una certa facilità implementare la trasmissione/ricezioni di files Ascii, aggiungendo magari un controllo **Xon-Xoff**, ma questo è già argomento per trattazioni prossime venture...

Il listato non presenta particolari difficoltà di interpretazione, fatta eccezione per qualche punto.

Anzitutto, con **On Break**... viene reso impossibile lo stop del programma, e questo per un motivo pratico: molte BBS (compresa la nostra) richiedono la presione di **Ctrl + C** per interrompere lo svolgimento in corso.

Eseguendo questa operazione, che per Amigabasic coincide con una richiesta di Stop, verrebbe però interrotto il programma, e quindi persa la comunicazione.

Redirigendo il Break alla label **"ferma"**, si fa in modo che a Ctrl + C corrisponda un invio del suo reale codice Ascii, ovvero **Chr$ (3)**.

Per uscire dal programma, si dovrà quindi necessariamente ricorrere all'opzione **Quit** del menu principale.

La scelta **"echo"**, poi, consente di vedere stampato sullo schermo quanto si digita da tastiera.

Utile nei dialoghi diretti con altri utenti, è però da disabilitare (**echo off**) quando ci si collega ad una BBS.

Per questa opzione, come anche per l'invio al modem dei comandi di chiamata e di inizializzazione, vengono adoperati dei comandi Hayes, dei quali avremo modo di parlare in seguito.

Inutile precisare che l'opzione **"telefona"** impone l'input di un numero di telefono, mentre **"ritelefona"** esegue automaticamente una chiamata all'ultimo numero impostato.

Per concludere, non resta che una nota per chi è alle prime armi: il ciclo **While...Wend** in cui è inglobata la fase di rice/trasmissione vera e propria, è praticamente un ciclo infinito, dato che -1 resterà sempre tale.

Un consiglio: provate a migliorare il programma, e magari fatecelo sapere.

Collegandovi a BBSystems, naturalmente...

PEN PAL, NON SOLO UN WORD PROCESSOR

Il pacchetto Pen Pal, oltre alla possibilità di inserire disegni nel testo, pemette di realizzare anche archivi consistenti grazie alla funzione di Database

di Luigi Callegari

Pen Pal della **SoftWood** (già nota per **File and Write**) riporta sulla confezione la scritta *Word Processor*, ma tale definizione è chiaramente restrittiva. Infatti si tratta più propriamente di un piccolo *pacchetto integrato*, contenente cioè non solo un ottimo **elaboratore di testi**, ma anche un archiviatore (**Database**) piuttosto evoluto, capace di svolgere operazioni complesse sui dati in archivio ed anche operazioni matematiche, *quasi* come uno **spreadsheet**. Inizialmente abbiamo uno schermo vuoto con soli due menu, **System** e **File**, dal quale si possono richiamare, anche contemporaneamente, gli altri programmi.

Menu System

Nello schermo iniziale, la funzione **Status** del menu *System* presenta statistiche sul sistema: Memoria Chip, Fast e Chunk più largo disponibile, disco inserito, spazio disponibile su disco, stato dei buffers ridimensionabili del programma.

La funzione **Preferences** consente di configurare tutti i vari sottoprogrammi che compongono Pen Pal tramite le voci: **Program, Document, DataBase, Text Print, Graphic Print**. In effetti tutti gli schermi che appaiono sono molto simili e consentono di stabilire parametri quali la directory (o meglio, la path) di memorizzazione del database o dei files grafici. Ad esempio, accedendo a Preferen-

Menu Edit

Qui sono contenute le funzioni di redazione del testo propriamente dette. Sono delle semplici operazioni di taglia/incolla, tipiche di tutti i programmi di videoscrittura.

La funzione **Undo** mostra un sottomenu che consente di annullare l'ultima operazione svolta sul testo.

Come al solito, non tutte le operazioni possono essere annullate, ma in questo caso la funzione del menu non è utilizzabile e compare scritta in modo *ghosted*. Undo è anche disattivato al primo click del mouse all'interno di una finestra di documento.

Cut ritaglia una porzione di testo o di grafico evidenziata dal documento e la inserisce nel budder **clipboard**.

La porzione di testo o di grafico viene poi cancellata dal documento. Questi dati possono essere reinseriti tramite la funzione **Paste**.

Copy funziona esattamente come **Cut**, fatto salvo che il testo o il grafico indicato viene copiato nel buffer di clipboard senza cancellarlo dal documento.

Paste visualizza un sottomenu che consente di inserire il testo od il grafico dalla clipboard (memorizzato con Cut o Copy) nel documento, al punto indicato dal cursore.

Clear rimuove il testo o la grafica selezionata dal documento, senza inserire nulla nel clipboard.

Select All visualizza un sottomenu che consente di selezionare tutto il testo o tutta la grafica di un documento in una volta sola.

New Page inserisce un codice che *forza* internamente Pen Pal ad andare all'inizio di una nuova pagina.

Insert Page # inserisce un numero di pagina automatico (aggiornato internamente) nel documento.

Insert Date inserisce una data automatica (aggiornata internamente) nel documento.

Menu Search

Qui sono raggruppate le funzioni che consentono di ricercare porzioni di testo alfanumerico e di sostituirlo.

Search / Replace fa comparire un requester che consente di immettere il testo da ricercare nel documento e, se necessario, il testo con il quale deve essere eventualmente sostituito.

Search **Next** consente di ricercare la successiva ricorrenza del testo usando il comando Search / Replace visto prima.

Replace consente di sostituire il testo trovato usando uno dei comandi precedenti. Search **&** Replace **Next** continua la ricerca di testo per sostituzione.

Search and Replace **All** ricerca e sostituisce automaticamente tutte le occorrenze del testo senza chiedere conferma.

PPExtras:Databases/Listings.FLR

Database Items: 100

	Dwelling	Location	Beds	Baths	Garage	Pool	Price
1	House	San Francisco	5	4	4	Yes	$5,000,000
2	House	San Clemente	5	4	4	Yes	$4,000,000
3	House	Malibu	4	3	3	Yes	$1,400,000
4	Penthouse	Century City	3	3	Y	Yes	$1,250,000
5	Beach House	Malibu	3	3	2	No	$1,000,000
6	House	Tucson	3	2	2	No	$900,000
7	Beach House	Santa Barbara	2	2	N	No	$750,000
8	Beach House	Santa Barbara	3	3	2	No	$600,000
9	House	Santa Barbara	5	3	3	Yes	$500,000
10	House	San Francisco	3	2	Y	Yes	$450,000
11	Duplex	Encino	4/4	3/3	2/2	No	$450,000
12	House	San Francisco	4	3	2	No	$400,000
13	Townhouse	Malibu	3	2	Y	No	$400,000
14	House	Pasadena	4	3	3	Yes	$400,000
15	House	Santa Barbara	4	3	3	No	$400,000
16	House	Santa Monica	4	3	3	No	$400,000
17	House	San Jose	3	2	Y	No	$350,000
18	House	Encino	4	3	3	No	$350,000

Go To Insertion Point sposta al punto del documento in cui si effettua l'inserimento.

Menu Process

Sono inserite varie funzioni di trattamento del testo e di controllo dell'ambiente di lavoro.

Spell mostra un sottomenu che consente lo spelling e la ricerca in dizionario (inglese) di una parola.

Forms Fill in visualizza un sottomenu utilizzabile per la creazione di formati personalizzati.

Statistics mostra una serie di statistiche sull'intero documento o sul paragrafo.

Page Setup mostra il requester di "Page Setup" che consente di definire le dimensioni della pagina.

Options mostra un sottomenu per configurare lo schermo di elaborazione del documento.

Ad esempio è possibile fare scomparire i gadget di scorrimento laterali ed il righello superiore.

Menu Graphics

Sono raggruppate le funzioni di gestione della grafica.

Import Picture attiva il processo di caricamento di un disegno in formato IFF da disco.

Text Flow mostra un sottomenu che consente di modificare le caratteristiche di un disegno selezionato.

Lock è usato per bloccare uno o più oggetti grafici in modo che non possano venire spostati accidentalmente durante l'edizione.

Unlock elimina il bloccaggio sugli elementi.

Group inserisce due o più oggetti grafici selezionati in un gruppo.

Ungroup rimuove tutti gli oggetti grafici raccolti in un gruppo.

Hide nasconde un grafico selezionato, che rimane nel documento ma non viene visualizzato.

ces / Document si possono personalizzare i dati inerenti ai documenti.

I requester richiedono ordinatamente: la path dove memorizzare per default i disegni (il default è **PPEXtras:Pictures/**, ovvero la directory Pictures del dischetto supplementare fornito col programma **PPExtras**), la directory dei documenti veri e propri (**Buffer:** per default), il documento da caricare come setup iniziale automaticamente (nessuno per default), dove reperire il vocabolario principale (PPExtras:**Speller/Main.DIC**), dove reperire il dizionario personalizzato dall'utente (PPExtras:Speller/**Main.DIC**) e dove reperire il dizionario comune (PPExtras: Speller /**Common.dic**).

In modo analogo, gli altri schermi di personalizzazione consentono di personalizzare e configurare l'ambiente di lavoro, anche se si dispone soltanto di un paio di drives, in modo soddisfacente e funzionale.

La funzione **Help!** di Preferences consente, tramite menu, di avere informazioni dettagliate su tutte le operazioni eseguibili dal programma.

Purtroppo i testi sono in inglese, ma chi conosce questa lingua potrà in questo modo accedere rapidamente ad informazioni sintetiche, ma chiare, sul funzionamento del programma.

Menu File

Tramite il menu **File** si possono caricare i moduli di funzionamento del programma veri e propri.

Le funzioni sono **New Document** e **New Database**, che caricano rispettivamente un nuovo documento ed un nuovo archivio. In questo caso vengono letti (dalla path definita con le funzioni di configurazione prima dette) i parametri di default per aprire un nuovo testo (documento) od un nuovo archivio (database).

Altrimenti si può utilizzare la funzione File / Open per aprire un documento od un archivio già esistenti.

In questo caso appare un requester che presenta (leggendo dalla path specificata tramite le funzioni di configurazione prima dette) i documenti od i database già inseriti.

Con le funzioni **Close All** e **Save All** si provvede, rispettivamente, a chiudere tutti i programmi attivati ed a salvare tutti gli aggiornamenti di tutti i programmi aperti in una volta sola.

Tramite l'opzione File / **Change Font List** è possibile stabilire una nuova fonte di caratteri.

Appare in questo caso un requester che richiede di specificare una directory contenente delle fonti di caratteri (simile, insomma, a *Fonts* del dischetto del Workbench).

Menu di Text

Aprendo un documento si attiva il programma **Text** di Pen Pal, che conta ben 9 menu: **System, File, Edit, Search, Process, Graphics, Text, Fonts, Size.** Il primo ha le stesse funzioni dello schermo iniziale, mentre gli altri sono specifici per l'elaboratore di testi.

Nel menu **File** sono contenute le funzioni di gestione dei files di documenti. Quasi tutte le opzioni funzionano aprendo una finestra, o meglio un requester, con vari gadget e bottoni che consentono di operare scelte.

L'opzione **Windows** visualizza un sottomenu che mostra i titoli di tutte le finestre aperte.

E' possibile attivare una di queste finestre scegliendo l'opzione da menu con il suo titolo.

New mostra un sottomenu con la scelta di definire un nuovo documento oppure un nuovo database, come visto prima.

Open apre un sottomenu che consente di aprire un documento od un database già esistente e stoccato sulla memoria di massa.

Close consente tramite un requester di chiudere un documento specifico oppure tutte le finestre correntemente aperte.

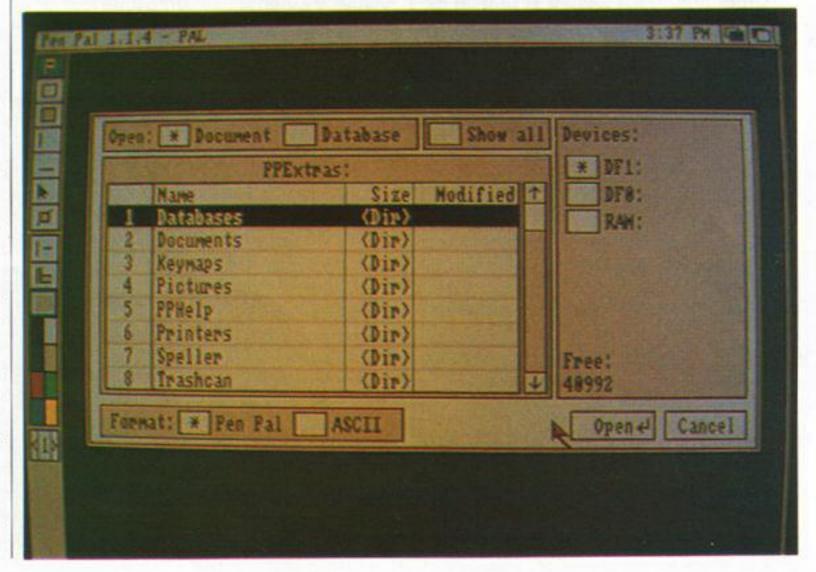
Save e Save As visualizzano un requester che consente di registrare il documento corrente o tutte le finestre correntemente aperte.

Save registra col nome specificato tramite l'ultimo uso di Save As, ciò implica ovviamente che queset'ultimo deve essere usato per primo, almeno una volta, prima di usare efficacemente il secondo.

Print mostra un requester che consente di stampare su carta un documento, una circolare (**mail** merge) o stampare un rapporto (**report** merge).

Rename visualizza un requester di file per alterare il nome del documento corrente.

Delete cancella il documento corrente, dopo una richiesta di conferma, dato che un documento cancellato non può più essere recuperato.

Quit è il comando che consente di terminare l'esecuzione di Pen Pal.

Show All mostra tutti i grafici nascosti nel documento.

Send to Back invia sul fondo dello stack di elaborazione di Pen Pal un grafico indicato.

Bring to Front porta un oggetto grafico selezionato in cima allo stack di elaborazione di Pen Pal.

Free Form consente, quando è attivata (affiancata sul menu dal simbolo di spunta), di spostare pixel per pixel un grafico sul documento.

Snap to 1/8 in. consente, quando è attivata, l'allineamento di oggetti grafici ad un *ottavo* di scatto sui marcatori orizzontali e verticali, per un perfetto allineamento.

Menu Text

Qui sono raggruppate le normali funzioni di controllo del testo.

Plain Text seleziona testo normale, disattivando tutti gli eventuali modi di scrittura speciali scelti con le funzioni seguenti: **Bold** scrive in grassetto, *Italic* in corsivo, Underline in sottolineato, Superscript in ascendente, Subscript in discendente.

Text Color fissa il colore del testo.

Text Alignment mostra un sottomenu che consente di cambiare i paragrafi selezionati in modo da allinearli ordinatamente.

Line Spacing mostra un sottomenu che consente di cambiare la spaziatura tra le linee del testo selezionato via mouse.

Menu Size

Contiene un solo comando che consente di stabilire la dimensione dei caratteri, indirettamente tramite scelta della fonte di testo, dato che ogni fonte non ha tutte le dimensioni possibili.

Ogni menu mostra un sottomenu con un gruppo di fonti.

Scegliendo una fonte, tutto il testo correntemente selezionato viene modificato alle dimensioni volute.

Se non è stato scelto (evidenziato) col mouse del testo allora la fonte di caratteri indicata diventa quella attuale, usata per la scrittura dei caratteri digitati da tastiera. Il simbolo di spunta nel menu indica la scelta corrente.

Menu Font

Consente di indicare il nome della fonte da usare.

Ciascun elemento del menu è quindi un nome di fonte che ha come sottomenu le dimensioni possibili.

Quando si sceglie una fonte qualunque da qui, la porzione di testo evidenziata diventa di quello stile.

Se non è stata scelta alcuna porzione di testo, lo stile e la dimensione indicati divengono quelli correntemente usati per scrivere.

Menu di Database

L'archiviatore di Pen Pal è piuttosto semplice, ma molto efficiente e perfettamente integrato col programma Text di videoscrittura. Vediamo le funzioni previste dai suoi menu, considerando che i menu **System, File** ed **Edit** sono identici a quelle viste per Text.

Menu Process

Vi sono contenute le funzioni di trattamento dei dati archiviati.

Sort List mostra il requester di scelta del sistema di riordinamento alfabetico.

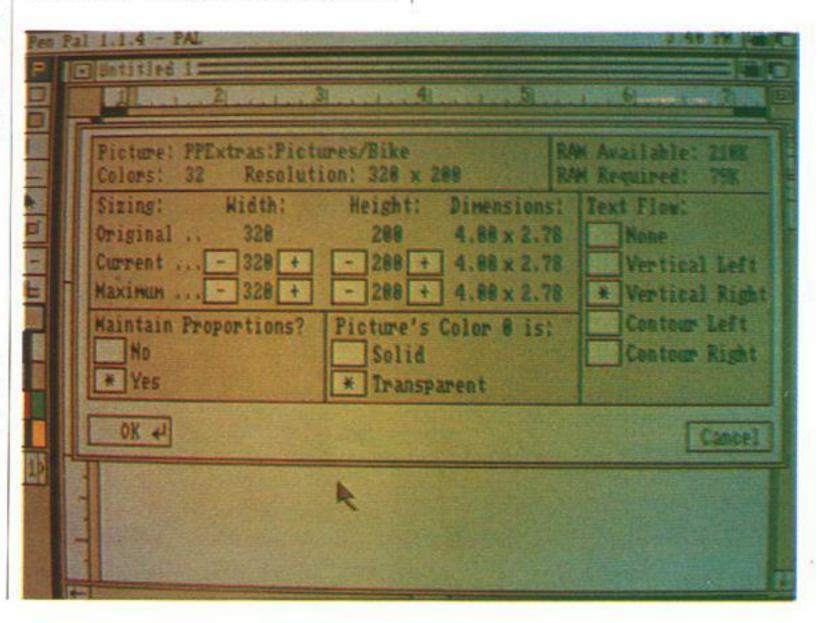
Search List mostra un sottomenu che consente di ricercare un elemento nell'archivio (con un metodo simile a quello visto per il Search di Text), oppure per localizzare i duplicati in colonne riordinate.

Show Rows mostra un sottomenu che consente di visualizzare o la lista di archiviazione principale, o una sottolista creata come risultato della ricerca.

Screen Totals mostra un sottomenu che consente di disattivare od attivare la visualizzazione dei dati totali sulla finestra del database.

Vol. 6 - n. 5

JONATHAN

DIMENSIONE AVVENTURA

L. 7000

TANZANIA

Safari in mongolfiera

INDIA

I tesori di Golconda

MONTAGNA

Trilogia messicana

SPECIALE

Avventura nei cieli

Jonathan mensile Maggio 1990 Anno VI N. 5 - Sped. Abb. Post. Gr. III/70 Tassa pagata per l'I.P.

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 27.000
Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 595.000
Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000
Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000
Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 7.812.000
Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 8.127.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000
Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000
Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000
Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000
Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000
Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000
Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000
PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.360.000
Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 1.675.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.095.000
Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 2.410.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000

Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000

Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000

Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000

Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000

Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D

1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64 - Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64 - Permette il collegamento a Videotel, P.G.E. e banche dati **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000

Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000

Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000

Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000

Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000

Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000

Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000

Nastro per stampante

Stampante MPS 1500C - L. 495.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000

Nastro a colori per stampante

MPS1500R - L.37.000

Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

LOMBARDIA

Milano

- AL RISPARMIO - V.LE MONZA 204
- BCS - VIA MONTEGANI 11
- BRAHA A. - VIA PIER CAPPONI 5
- E.D.S. - C.SO PORTA TICINESE 4
- FAREF - VIA A. VOLTA 21
- FLOPPERIA - V.LE MONTENERO 31
- GBC - VIA CANTONI 7 - VIA PETRELLA 6
- GIGLIONI - V.LE LUIGI STURZO 45
- L'UFFICIO 2000 - VIA RIPAMONTI 213
- LOGITEK - VIA GOLGI 60
- LU - MEN - VIA SANTA MONICA 3
- MARCUCCI - VIA F.LLI BRONZETTI 37
- MELCHIONI - VIA P. COLLETTA 37
- MESSAGGERIE MUSICALI - GALLERIA DEL CORSO 2
- NEWEL - VIA MAC MAHON 75
- PANCOMMERZ ITALIA - VIA PADOVA 1
- SUPERGAMES - VIA VITRUVIO 38
- 68000 E DINTORNI - VIA WASHINGTON 91

Provincia di Milano

- GINO FERRARI CENTRO HI-FI - VIA MADRE CABRINI 44 - S. ANG. LODIGIANO
- F.LLI GALIMBERTI - VIA NAZIONALE DEI GIOVI 28/36 - BARLASSINA
- TECNOLUX - VIA PIETRO NENNI 5 - BERNATE TICINO
- OGGIONI & C. - VIA DANTE CESANA 27 - CARATE BRIANZA
- AL RISPARMIO - VIA U. GIORDANO 57 - CINISELLO BALSAMO
- GBC - V.LE MATTEOTTI 66 - CINISELLO BALSAMO
- CASA DELLA MUSICA - VIA INDIPENDENZA 21 - COLOGNO MONZESE
- PENATI - VIA VERDI 28/30 - CORBETTA
- EPM SYSTEM - V.LE ITALIA 12 - CORSICO
- P.G. OSTELLARI - VIA MILANO 300 - DESIO
- CENTRO COMPUTER PANDOLFI - VIA CORRIDONI 18 - LEGNANO
- COMPUTEAM - VIA VECELLIO 41 - LISSONE
- M.B.M. - C.SO ROMA 112 - LODI
- L'AMICO DEL COMPUTER - VIA CASTELLINI 27 - MELEGNANO
- BIT 84 - VIA ITALIA 4 - MONZA
- IL CURSORE - VIA CAMPO DEI FIORI 35 - NOVATE MIL.
- I.C.O. - VIA DEI TIGLI 14 - OPERA
- R & C ELGRA - VIA SAN MARTINO 13 - PALAZZOLO MIL.
- ESSEGIEMME SISTEMI SAS - VIA DE AMICIS 24 - RHO
- TECNO - CENTRO - VIA BARACCA 2 - SEREGNO
- NIWA HARD&SOFT - VIA B. BUOZZI 94 - SESTO SAN GIOV.
- COMPUTER SHOP - VIA CONFALONIERI 35 - VILLASANTA
- ACTE - VIA B. CREMIGNANI 13 - VIMERCATE
- IL COMPUTER SERVICE SHOP - VIA PADANA SUPERIORE 197 - VIMODRONE

Bergamo

- D.R.B. - VIA BORGO PALAZZO 65
- TINTORI ENRICO &C. - VIA BROSETA 1
- VIDEO IMMAGINE - VIA CARDUCCI c/o CITTA' DI MERCATO

Provincia di Bergamo

BERTULEZZI GIOVANNI - VIA FANTONI 48 - ALZANO LOMBARDO
- COMPUTER SHOP - VIA VITTORIO VENETO 9 - CAPRIATE SAN GERVASIO
- B.M.R. - VIA BUTTARO 4/T - DALMINE
- MEGABYTE 2 - VIA ROMA 61/A - GRUMELLO
- OTTICO OPTOMETRISTA ROVETTA - P.ZZA GARIBALDI 6 - LOVERE
- COMPUTER POINT - VIA LANTIERI 52 - SARNICO
- A.B. INFORMATICA - STRADA STATALE CREMASCA 66 - URGNANO

Brescia

- MASTER INFORMATICA - VIA F.LLI UGONI 10/B

PROVINCIA DI BRESCIA

- MISTER BIT - VIA MAZZINI 70 - BRENO
- CAVALLI PIETRO - VIA 10 GIORNATE 14 BIS - CASTREZZATO
- VIETTI GIUSEPPE - VIA MILANO 1/B - CHIARI
- MEGABYTE - P.ZZA MALUEZZI 14 - DESENZANO DEL GARDA
- BARESI RINO &C. - VIA XX SETTEMBRE 7 - GHEDI
- INFO CAM - VIA PROVINCIALE 3 - GRATACASOLO
- PAC-LAND" di GARDONI - CENTRO COM.le - LA CASA DI MARGHERITA D'ESTE - VIA GIORGIONI 21

Como

- IL COMPUTER - VIA INDIPENDENZA 90
- 2M ELETTRONICA - VIA SACCO 3

Provincia di Como

- ELTRON - VIA IV NOVEMBRE 1 - BARZANO'
- DATA FOUND - VIA A. VOLTA 4 - ERBA
- CIMA ELETTRONICA - VIA L. DA VINCI 7 - LECCO
- FUMAGALLI - VIA CAIROLI 48 - LECCO
- RIGHI ELETTRONICA - VIA G. LEOPARDI 26 - OLGIATE COMASCO

Cremona

- MONDO COMPUTER - VIA GIUSEPPINA 11/B
- PRISMA - VIA BUOSO DA DOVARA 8
- TELCO - P.ZZA MARCONI 2/A

Provincia di Cremona

- ELCOM - VIA IV NOVEMBRE 56/58 - CREMA
- EUROELETTRONICA - VIA XX SETTEMBRE 92/A - CREMA

Mantova

- COMPUTER CANOSSA - GAL. FERRI 7
- 32 BIT - VIA C. BATTISTI 14
- ELET. di BASSO - V.LE RISORGIMENTO 69

Provincia di Mantova

- CLICK - ON COMPUTER - S.S. GOITESE 168 - GOITO

Pavia

- POLIWARE - C.SO C. ALBERTO 76
- SENNA GIANFRANCO - VIA CALCHI 5

Provincia di Pavia

- A. FERRARI - C.SO CAVOUR 57 - MORTARA
- LOGICA MAINT - V.LE M.TE GRAPPA 32 - VIGEVANO
- M. VISENTIN - C.SO V. EMANUELE 76 - VIGEVANO

Sondrio

- CIPOLLA MAURO - VIA TREMOGGE 25

Provincia di Sondrio

- FOTONOVA - VIA VALERIANA 1 - S.PIETRO DI BERBENNO

Varese

- ELLE - EFFE - VIA GOLDONI 35
- IL C.TRO ELET. - VIA MORAZZONE 2
- SUPERGAMES - VIA CARROBBIO 13

Provincia di Varese

- BUSTO BIT - VIA GAVINANA 17 - BUSTO A.
- MASTER PIX - VIA S.MICHELE 3 - BUSTO A.
- PUNTO UFFICIO - VIA R.SANZIO 8 - GALLARATE
- GRANDI MAGAZZINI BOSSI - VIA CLERICI 196 - GERENZANO
- J.A.C. - C.so MATTEOTTI 38 - SESTO C.

PIEMONTE

Alessandria

- BIT MICRO - VIA MAZZINI 102
- SERV. INFOR. - VIA ALESSANDRO III 47

Provincia di Alessandria

- SONY ITALIANA - VIA G. MANARA 7 - CASALE MONFERRATO
- SGE ELETTRONICA - VIA BANDELLO 19 - TORTONA
- COMPUTER TEMPLE - VIA F. CAVALLOTTI 13 - VALENZA

Asti

- ASTI GAMES - C.SO ALFIERI 26
- RECORD - C.SO ALFIERI 166/3 (Galleria Argenta)

Cuneo

- ROSSI COMPUTERS - C.SO NIZZA 42

Provincia di Cuneo

- PUNTO BIT - C.SO LANGHE 26/C - ALBA
- BOSETTI - VIA ROMA 149 - FOSSANO
- COMPUTERLAND - VIA MAZZINI 30/32 - SALUZZO

Novara

- PROGRAMMA 3 - V.LE BUONARROTI 8
- PUNTO VIDEO - C.so RISORGIMENTO 39/B

Provincia di Novara

- COMPUTER - VIA MONTE ZEDA 4 - ARONA
- ALL COMPUTER - C.SO GARIBALDI 106 - BORGOMANERO
- S.P.A. - C.SO DISSEGNA 21/BIS - DOMODOSSOLA
- ELLIOTT COMPUTER SHOP - VIA DON MINZONI 32 - INTRA
- TRISCONI VALERIA - VIA MAZZINI 90 - OMEGNA

Torino

- ABA ELETTRONICA - VIA C. FOSSATI 5/P
- ALEX COMPUTER E GIOCHI - C.SO FRANCIA 333/4
- COMPUTER HOME - VIA SAN DONATO 46/D
- COMPUTING NEW - VIA M. POLO 40/E
- C.D.M. ELETTR. - VIA MAROCHETTI 17
- DE BUG - C.SO V. EMANUELE II 22
- DESME UNIVERSAL - VIA S.SECONDO 95
- FDS ALTERIO - VIA BORGARO 86/D
- IL COMPUTER - VIA N. FABRIZI 126
- MICRONTEL - C.SO D. degli ABRUZZI 28
- PLAY GAMES SHOP - VIA C. ALBERTO 39/E
- RADIO TV MIRAFIORI - C.SO UNIONE SOVIETICA 381
- SMT ELETTRONICA - VIA BIBIANA 83/bis

Provincia di Torino

- PAUL E CHICO VIDEOSOUND - VIA V.EMANUELE 52 - CHIERI
- BIT INFORMATICA - VIA V. EMANUELE 154 - CIRIE'
- HI - FI CLUB - C.SO FRANCIA 92C - COLLEGNO
- MISTER PERSONAL - VIA CATTANEO 52 - FAVRIA
- I.C.S. - VIA TORINO 73 - IVREA
- DAG - VIA I MAGGIO 40 - LUSERNA S. GIOVANNI
- EUREX - C.SO INDIPENDENZA 5 - RIVAROLO CANAVESE
- DIAM INFORMATICA - C.SO FRANCIA 146 bis - RIVOLI
- FULLINFORMATICA - VIA V.VENETO 25 - RIVOLI
- GAMMA COMPUTER - VIA CAVOUR 3A-3B - SET.TORINESE

Vercelli

- ELETTROGAMMA - C.SO BORMIDA 27 ang. V.Montanara
- ELETTRONICA - STRADA TORINO 15

Provincia di Vercelli

- C.S.I. TEOREMA - VIA LOSANA 9 - BIELLA
- SIGEST - VIA BERTODANO 8 - BIELLA
- REMONDINO FRANCO - VIA ROMA 5 - BORGOSESIA
- FOTOSTUDIO TREVISAN - VIA XXV APRILE 24/B - COSSATO
- STUDIO FOTOGRAFICO IMARISIO - P.ZZA M. LIBERTA' 7 - TRINO

VENETO

Belluno

- UP TO DATE - VIA V. VENETO 43

Provincia di Belluno

- GUERRA COMPUTERS - V.LE MAZZINI 10/A - FELTRE

Padova

- BIT SHOP - VIA CAIROLI 11
- COMPUMANIA - VIA T. CAMPOSANPIERO 37
- D.P.R. DE PRATO R. - V.LO LOMBARDO 4
- G.F. MARCATO - VIA MADONNA DELLA SALUTE 51/53
- SARTO COMPUTER - VIA ARMISTIZIO 79

Provincia di Padova

- COMPUTER SERVICE - BORGO TREVISO 150 - CITTADELLA

Treviso

- BIT 2000 - VIA BRANDOLINI D'ADDA 14
- GUERRA EGIDIO & C. - V.LE CAIROLI 95

Provincia di Treviso

- DE MARIN COMPUTERS - VIA MATTEOTTI 142 - CONEGLIANO
- SIDESTREET - VIA SALVO D'ACQUISTO 8 - MONTEBELLUNA
- FALCON ELETTROAUDIOVIDEO - VIA TERRAGGIO 116 - PREGANZIOL

Venezia

- GUERRA EGIDIO & C. - VIA BISSUOLA 20/A - MESTRE
- TELERADIO FUGA - SAN MARCO 3457

Provincia di Venezia

- GUERRA EGIDIO & C. - VIA VIZZOTTO 29 - SAN DONA' DI PIAVE
- REBEL - VIA F. CRISPI 10 - SAN DONA' DI PIAVE

Verona

- CASA DELLA RADIO - VIA CAIROLI 10
- TELESAT - VIA VASCO DE GAMA 8

Provincia di Verona

- UBER - CP 0363(RAG SOC. DERTA) - VIA MASCAGNI 31 - CASTEL D'AZZANO
- FERRARIN - VIA DEI MASSARI 10 - LEGNAGO
- COMPUTERS CENTER - VIA CANTORE 26 - VILLAFRANCA

Vicenza

- ELET. BISELLO - V.LE TRIESTE 427/429
- SCALCHI MARKET - VIA CA' BALBI 139

Provincia di Vicenza

- SCHIAVOTTO - VIA ZANELLA 21 - CAVAZZALE
- GUERRA E. & C. - V.LE DELLE INDUSTRIE - MONTECCHIO MAGGIORE

FRIULI VENEZIA GIULIA

Gorizia

- E.CO. ELETTRONICA - VIA F.LLI COSSAR 23

Trieste

- AVANZO GIACOMO - P.ZZA CAVANA 7
- COMPUTER SHOP - VIA P. RETI 6
- COMPUTIGI - VIA XX SETTEMBRE 51
- CTI - VIA PASCOLI 4

Udine

- MOFERT 2 - VIA LEOPARDI 21
- R.T. SISTEM UDINE - VIA L. DA VINCI 99

Provincia di Udine

- IL PUNTO ELETTRONICO - VIA VENDRAMIN 184 - LATISANA
- IDRENO MATTIUSSI &C. - VIA LICINIANA 58 - MARTIGNACCO

TRENTINO ALTO ADIGE

Bolzano

- COMPUTER POINT - VIA ROMA 82/A
- MATTEUCCI PRESTIGE - VIA MUSEO 54

Provincia di Bolzano

- RADIO MAIR-ELECTRO - VIA CENTRALE 70 - BRUNICO
- ELECTRO RADIO HENDRICH - VIA DELLE CORSE 106 - MERANO
- ERICH KONTSCHIEDER - PORTICI 313 - MERANO
- ELECTRO TAPPEINER - P.ZZA PRINCIPALE 90 - SILANDRO

Trento

- CRONST - VIA G. GALILEI 25

Provincia di Trento

• AL RISPARMIO - C.SO VERONA 138 - ROVERETO

LIGURIA
Genova
• ABM COMPUTER - P.ZZA DE FERRARI 24 rosso
• CAPRIOTTI G. - IA MAMIANI 4r - SAMPIERDARENA
• C.tro ELET. - VIA CHIARAVAGNA 10 R - VIA SESTRI 69R
• COM.le SOTTORIPA - VIA SOTTORIPA 115/117
• FOTOMONDIAL - VIA DEL CAMPO 3-5-9-11-13 r
• LA NASCENTE - VIA SAN LUCA 4/1
• PLAY TIME - VIA GRAMSCI 3/5/7 rosso
• RAPPR-EL - VIA BORGORATTI 23 R
Imperia
• CASTELLINO - VIA BELGRANO 44
Provincia di Imperia
• CENTRO HI-FI VIDEO - VIA DELLA REPUBBLICA 38 -SANREMO
• CASTELLINO - VIA GENOVA 48 - VENTIMIGLIA
La Spezia
• I.L. ELETTRONICA - VIA V. VENETO 123
Provincia di La Spezia
• I.L. ELETTRONICA - VIA AURELIA 299 - FORNOLA DI VEZZANO
Savona
• CASTELLINO - C.SO TARDY E BENECH 101
Provincia di Savona
• CELESIA ENZA - VIA GARIBALDI 144 - LOANO

EMILIA
Bologna
• EUROELETTRICA - VIA RANZANI 13/2
• MINNELLA ALTA FEDELTA' - VIA MAZZINI 146/2
• MORINI & FEDERICI - VIA MARCONI 28/C
• STERLINO - VIA MURRI 73/75
Provincia di Bologna
• S.C. COMPUTERS - VIA E. FERMI 4 - CASTEL SAN PIETRO
• S.P.E. INFORMATICA - VIA DI MEZZO PONENTE 385 - CREVALCORE
• ARCHIMEDE SISTEMI - VIA EMILIA 124 - S. LAZZARO DI SAVENA
Modena
• CO - EL - VIA CESARI 7
• ORSA MAGGIORE - P.ZZA MATTEOTTI 20
• VIDEO VAL WILLY COMPUTERS - VIA CANALETTO 223
Provincia di Modena
• NEW MEDIA SYSTEM - VIA ROMA 281 - SOLIERA
Parma
• BABARELLI G. - VIA B. PARENTE 14/A/B
Provincia di Parma
• PONGOLINI - VIA CAVOUR 32 - FIDENZA
Piacenza
• COMPUTER LINE - VIA G. CARDUCCI 4
• DELTA COMPUTER - VIA M. DELLA RESISTENZA 15/G

TEGGIO EMILIA
• COMPUTERLINE - VIA SAN ROCCO 10/C
• POOL SHOP - VIA EMILIA S. STEFANO 9/C
Provincia di Reggio Emilia
• MACCHIONI - VIA STATALE 467 - CASALGRANDE
ROMAGNA
Ferrara
• BUSINESS POINT - VIA CARLO MAYER 85
Forlì
• COMPUTER VIDEO CENTER - VIA CAMPO DI MARTE 122
Provincia di Forlì
• TOP BIT - VIA VENETO 12 - FORLIMPOPOLI
• COMPUTER HOUSE - V.LE TRIPOLI 193/D - RIMINI
• EASY COMPUTER - VIA LAGOMAGGIO 50 - RIMINI

REPUBBLICA S. MARINO
Ravenna
• COMPUTER HOUSE - VIA TRIESTE 134
Provincia di Ravenna
• ARGNANI - P.ZZA DELLA LIBERTA' 5/A - FAENZA
• ELECTRON INFORMATICA - VIA F.LLI CORTESI 17 - LUGO
• P.L.Z. INFORMATICA - P.ZZA SERCOGNANI 6 - FAENZA

TOSCANA
Arezzo
• DELTA SYSTEM - VIA PIAVE 13
Firenze
• ATEMA - VIA BENEDETTO MARCELLO 1a-1b
• ELETTRONICA CENTOSTELLE - VIA CENTO STELLE 5/a-b
• HELP COMPUTER - VIA DEGLI ARTISTI 15-A
• TELEINFORMATICA TOSCANA -VIA BRONZINO 36
Provincia di Firenze
• WAR GAMES - VIA R. SANZIO 126/A - EMPOLI
• NEW EVM COMPUTER - VIA DEGLI INNOCENTI 2 - FIGLINE VALDARNO
• C.tro INFOR. - VIA ZNOJMO 41 - PONTASSIEVE
• COSCI F.LLI - VIA ROMA 26 - PRATO
• BARBAGLI C. ELET. - VIA F. BONI 80 - PRATO
Grosseto
• COMPUTER SERVICE - VIA DELL'UNIONE 7
Livorno
• ETA BETA - VIA SAN FRANCESCO 30
• FUTURA 2 - VIA CAMBINI 19
Provincia di Livorno
• PUNTO ROSSO - VIA BARONTINI 28 - PIOMBINO
Provincia di Lucca
• IL COMPUTER - V.LE COLOMBO 216 - LIDO DI CAMAIORE
• SANTI VITTORIO - VIA ROMA 23 - S. ROMANO GARFAGNANA
• TOP GAMES - VIA S. ANDREA 122 - VIAREGGIO
Massa
• EURO COMPUTER - P.ZZA G. BERTAGNINI 4
Carrara
• RADIO LUCONI - VIA ROMA 24/B
Pisa
• ELECTRONIC SERVICE - VIA DELLA VECCHIA TRANVIA 10
• PUCCINI S.- CP 1199 (RAG.SOC. MAREX) - VIA C.CAMMEO 64
• TONY HI-FI - VIA CARDUCCI
Provincia di Pisa
• M.C. INFORMATICA - VIA DEL CHIESINO 4 - PONTEDERA (PI)
Pistoia
• ELECTRONIC SHOP - VIA DEGLI SCALZI 3
Provincia di Pistoia
• ZANNI &C. - C.SO ROMA 45 - MONTECATINI T.

Siena
• R. BROGI - P.ZZA GRAMSCI 28
• VIDEO MOVIE - VIA GARIBALDI 17
Provincia di Siena
• ELETTRONICA di BIFOLCHI - VIA DI GRACCIANO NEL CORSO 111 - MONTEPULCIANO

LAZIO
• CENTRO INF. - D.R.R. srl - TEL. 06-5565672

UMBRIA
Perugia
• MIGLIORATI - VIA S. ERCOLANO 3-10
Provincia di Perugia
• COMPUTER STUDIO'S - VIA IV NOVEMBRE 18/A - BASTIA UMBRA
• WARE - VIA DEI CASCERI 31 - CITTA' DI CASTELLO
Terni
• CGS SOFTWARE HOUSE - VIA DONIZETTI 71/A

BASILICATA
Matera
• G. GAUDIANO ELECTRONICS - VIA ROMA ang. XX SETTEMBRE 1

PUGLIA
Bari
• ARTEL - VIA GUIDO D'ORSO 9
• COMPUTER'S ARTS - V.LE MEUCCI 12/B
• PAULICELLI S. & F. - VIA FANELLI 231/C
Provincia di Bari
• F. FAGGELLA - C.SO GARIBALDI 15 - BARLETTA
• G.FAGGELLA - P.ZZA D'ARAGONA 62A - BARLETTA
• LONUZZO G. - VIA NIZZA 21 - CASTELLANA
• TECNOUFF. - VIA RICASOLI 54 - MONOPOLI
• TANGORRA N. - C.SO V.EMANUELE 130/B - TRIGGIANO
Brindisi
• MARANGI E NICCOLI - VIA PROV. SAN VITO 165
Provincia di Brindisi
• MILONE G. - VIA S.F. D'ASSISI 219 - FRANCAVILLA FONTANA
Foggia
• BOTTICELLI G. - VIA SAV POLLICE 2
• E.C.I. COMPUTER - VIA ISONZO 28
• LA TORRE - V.LE MICHELANGELO 185
Provincia di Foggia
• IL DISCOBOLO - VIA T. SOLIS 15 - SAN SEVERO
Lecce
• BIT - VIA 95 REGG.NTO FANTERIA 87/89
Provincia di Lecce
• TECNO UFFICIO - P.ZZA GIOVANNI XXIII 10 - GALLIPOLI
• CEDOK INFORMATICA - VIA UMBERTO I 116 - TRICASE
Taranto
• ELETTROJOLLY C.tro - VIA DE CESARE 13
• TEA - TEC. ELET. AV. - VIA R. ELENA 101

CAMPANIA
Provincia di Avellino
• FLIP FLOP - VIA APPIA 68 - ATRIPALDA
Benevento
• E.CO. INF. - VIA PEPICELLI 21/25
Caserta
• ENTRY POINT - VIA COLOMBO 31
• O.P.C. - VIA G. M. BOSCO 24
Provincia di Caserta
• M.P. COMPUTER - VIA NAPOLI 30 - MADDALONI
• DAMIANO - C.SO V. EMANUELE 23 -ORTA DI ATELLA
• FUSCO B. - VIA NAPOLI 24 - VAIRANO PATERNORA (FRAZ. VAIRANO SCALO)
• LINEA CONTABILE - VIA OSPEDALE 72/76 - SESSA A. (CE)
Napoli
• BABY TOYS - VIA CISTERNA DELL'OLIO 5/BIS
• CASA MUSICALE RUGGIERO - P.ZZA GARIBALDI 74 (INT. STAZ. F.F. S.S.)
• C.tro ELET. CAMPANO - VIA EPOMEO 121
• CI.AN - GALLERIA VANVITELLI 32
• CINE NAPOLI - VIA S. LUCIA 93/95
• DARVIN - CALATA SAN MARCO 26
• GIANCAR 2 - P.ZZA GARIBALDI 37
• ODORINO - L.GO LALA 22 A-B
• R 2 - VIA F. CILEA 285
• SAGMAR - VIA S. LUCIA 140
• TOP VIDEO - TOP COMPUTER - VIA S. ANNA DEI LOMBARDI 12
• VIDEOFOTOMARKET - VIA S. BRIGIDA 19
Provincia di Napoli
• ELECTRONIC DAY - VIA DELLE PUGLIE 17 - CASORIA
• TUFANO - S.S. SANNITICA 87 KM 7 - CASORIA
• SOF SUD - V.LE EUROPA 59 - CASTEL/MARE DI STABIA
• ELETTRONICA 2000 - C.SO DURANTE 40 - FRATTAMAGGIORE
• SPADARO - VIA ROMANI 93 - MADONNA DELL'ARCO
• GATEWAY - VIA NAPOLI 68 - MUGNANO
• VISPINI & DI VUOLO - VIA A.ROSSI 4 - POMPEI
• SPY CASH & CARRY - P.ZZA ARENELLA 6/A - NAPOLI
• NUOVA INFORMATICA SHOP - VIA LIBERTA' 185/191 - PORTICI
• BASIC COMPUTER - C.SO GARIBALDI 34 - POZZUOLI
• V.C. - C.SO SECONDIGLIANO 562/B - SECONDIGLIANO
• F. ELETTRONICA - VIA SARNO 102 - STRIANO
• TECNO - VIA V. VENETO 48 - TORRE DEL GRECO
Salerno
• COMPUMARKET - VIA BELVEDERE 35
• COMPUTER MARKET - C.SO VITTORIO EMANUELE 23
Provincia di Salerno
• KING COMPUTER - VIA OLEVANO 56 - BATTIPAGLIA
• DIMER POINT - V.LE AMENDOLA 36 - EBOLI
• IACUZIO F. - VIA MUNICIPIO 14 - MERCATO SAN SEVERINO
• COMPUTER SERVICE - VIA L.DA VINCI 81 - SCAFATI

CALABRIA
Catanzaro
• C. & G. COMPUTER - VIA F. ACRI 28
• PAONE S. & F. - VIA F. ACRI 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - VIA BOLOGNA (L.GO OSPEDALE) - CROTONE
• RIOLO F.LLI - VIA VENEZIA 1/7 - CROTONE
• ING. FUSTO S. - C.SO NICOTERA 99 - LAMEZIA TERME
Cosenza
• MAISON DE L'INFORMATIQUE - VIA PASQUALE ROSSI 34/C
• SIRANGELO COMP. - VIA N. PARISIO 25
Provincia di Cosenza
• HI-FI ALFANO G. - VIA BALDACCHINI 109 - AMANTIA
• ELIGIO ANNICCHIARICO &C. - VIA ROMA 21 - CASTROVILLARI
• ALFA COMPUTER - VIA NAZIONALE 341/A - CORIGLIANO SCALO
REGGIO CALABRIA
• CONTROL SYSTEM - VIA S.F DA PAOLA 49 D
• SYSTEM HOU. - VIA FIUME ang. PALESTINO 1
Provincia di Reggio Calabria
• COMPUTER SHOP - V.LE MATTEOTTI 36/38 - LOCRI
• PICIEFFE - C.SO F. S. ALESSIO 19 - TAURIANOVA
SICILIA
• CENTRO INF. - ITALSOFT SRL - TEL. 0935-696090

Leggo VR perché mi dà la rotta

Il lettore di VR è giovane, dinamico, creativo. Di cultura e reddito superiore alla media, possiede spesso più di un videoregistratore, oltre all'impianto hi-fi e al computer: nel tempo libero, non rinuncia ai viaggi in Italia e all'estero, e a cinema, teatro e spettacoli sportivi in genere. Usa il videoregistratore non solo per i programmi tv o preincisi, ma anche per riprendere i momenti felici in famiglia, per creare una videoteca personale.
E tu, che tipo di lettore sei?

VR
VIDEOREGISTRARE

IMMAGINE MARSILII